Facultad de Ingeniería y Computación

Escuela Profesional de Ingeniería de Telecomunicaciones

“Mejora para la autenticación de IPv6 utilizando el protocolo KERBEROS”

Presentado por:

Kenneth Urquizo Enriquez

Para Optar por el Título Profesional de:

INGENIERO DE TELECOMUNICACIONES

Orientador: “Mag. Julio Omar Santisteban Pablo”

Arequipa, Septiembre de 2017

PROGRAMA PROFESIONAL DE INGENIERÍA DE TELECOMUNICACIONES

MEJORA PARA LA AUTENTICACIÓN DE IPV6 UTILIZANDO EL PROTOCOLO KERBEROS

Autor: Kenneth Urquizo Enriquez

September, 2017

"El experimentador que no sabe lo que está buscando no comprenderá lo que encuentra"

Claude Bernard, 1813-1878

Índice general

Índice de figuras

Abstract

This document presents a protocol which improves the IPv6 authentication, allowing different nodes to exchange packets in a safe way. Both the IPv6 and the Kerberos protocol are combined to create a new IPv6 protocol. The new protocol identifies the nodes by using different session and service tickets, it also detects any intruder (man in the middle) and if it is the case stops the communication. Using different session and services keys the integrity of the packets is assured and the identity of the nodes is verified. The methodology in this work can be applied to different networks, even scalable networks and can be implemented into other models of IPv6 communications.

Resumen

El incremento en el número de usuarios de internet nos ha obligado a realizar cambios y buscar mejoras en los protocolos de internet (IPs). Las grandes demandas, junto con el crecimiento del internet, trajo como resultado la creación del protocolo de internet de versión 4 o IPv4. Sin embargo debido al inminente incremento del número usuarios, se vio en la necesidad de crear un nuevo protocolo que supere las capacidades del IPv4, el protocolo de internet de versión 6 o IPv6.

Lamentablemente el IPv6 todavia sigue teniendo puntos débiles, tales como nuevos tipos de ataques a las redes. Es por ello que es necesario realizar un estudio completo de la forma en que IPv6 trabaja, para que de esta forma se pueda plantear nuevas soluciones y mejoras a esta tecnología.

Este documento presenta un protocolo que mejora la autenticación IPv6, permitiendo a diferentes nodos intercambiar paquetes de forma segura. Tanto el IPv6 como el protocolo Kerberos se combinan para crear un nuevo protocolo IPv6. El nuevo protocolo identifica los nodos utilizando diferentes tickets de sesión y servicio, también detecta cualquier intruso (man in the middle) y si es el caso interrumpe la comunicación. Utilizando diferentes claves de sesión y servicios, se garantiza la integridad de los paquetes y se verifica la identidad de los nodos. La metodología de este trabajo se puede aplicar a diferentes redes, incluso redes escalables y puede ser implementada en otros modelos de comunicaciones IPv6.

1 Introducción

Uno de los temas más importantes y más estudiados en las redes de Telecomunicaciones, es seguridad en las redes de datos. Desde los últimos años, muchas empresas han dedicado mucho tiempo a la investigación y desarrollo de nuevas y mejores formas para garantizar mayores niveles de seguridad en las redes, y ofrecer seguridad a todos los usuarios de internet. Seguridad que busca ser lo más transparente posible para los usuarios y a prueba de errores y ataques.

Si queremos adentrarnos en este tema y con ello proponer nuevas soluciones, es necesario realizar un estudio de los elementos más básicos que intervienen en las comunicaciones basadas en internet, para de esta forma solucionar o mejorar algunos de los puntos débiles en las comunicaciones. Estos elementos básicos son la clave, dado que el ámbito de la seguridad en las redes es un tema muy amplio. Entre los elementos básicos, y por no decir el más importante, se tiene a los protocolos de internet o IP, de los cuales hablaremos brevemente a continuación.

El protocolo de internet es el elemento más importante que garantiza una comunicación basada en internet, y debido a ello todos los estudios que apuntan a la seguridad en las redes realizan estudios sobre la IP. Gracias a los constantes estudios en los protocolos de internet, se han desarrollado notables avances en la seguridad de internet, logrando mayor rendimiento, confiabilidad, compatibilidad, mayor velocidad en las comunicaciones, cubrir un mayor número de usuarios, y muchos otros avances, todos estos sobre la IP.

En la actualidad existe una nueva versión de IP, con la que se está trabajando día a día, estamos hablando del protocolo de internet versión 6 o IPv6. El IPv6, como tecnología de la actualidad, presenta numerosas ventajas sobre su predecesor, ventajas que garantizan mayor seguridad para sus usuarios de internet, y dado que el IPv6 es la tecnología con la que se está trabajando hoy en día y al mismo tiempo es la tecnología a la que todo equipo migrará en los próximos años, es necesario centrar los estudios en buscar nuevas formas de mejorar IPv6, y hacer de esta tecnología la fuente que garantizará las comunicaciones en los próximos años. Desde ahora indicaremos que incluso IPv6 no garantiza una completa seguridad en internet, dado que existen numerosos ataques a las redes que incluso IPv6 no puede detectar.

Este proyecto busca mediante una recopilación de los aspectos más relevantes del IPv6, orientados a la autenticación. Presentar los puntos a mejorar en la autenticación de IPv6, para proponer una mejora en la forma de autenticación, basada en la autenticación de IPv6 ya existente. Esta nueva propuesta, la cual se denominara "IPv6 Kerberos" o de forma resumida "IPv6K", solucionará uno de los tantos ataques en IPv6, el cual es el "el hombre en el medio" (Man in the Middle), aumentando así su nivel de seguridad en la autenticación con los usuarios que se comunican con IPv6.

El presente trabajo solo busca solucionar el ataque de "el hombre en el medio" en IPv6. Con ello adelantaremos que la propuesta a desarrollar solo mejora la autenticación de IPv6 y no hace a IPv6 un protocolo a prueba de todos los ataques, dado que esta tarea demandaría mayor tiempo y más estudios [IPv00]. Se busca que este proyecto sirva de ayuda para mejorar los aspectos faltantes de IPv6, para hacer de IPv6 un protocolo más seguro y a prueba de todos los ataques a las redes de internet.

1.1. Motivación y contexto

Es conocido por todos que las comunicaciones de hoy en día, pese a los grandes avances tecnológicos, no son capaces de ofrecer una integridad completa a sus usuarios. Esto se debe a que hoy en día existen constantes ataques a las redes de comunicaciones. Es por ello que la tarea de buscar mayores y mejores avances en los protocolos de internet, es la base para garantizar mayor integridad en las comunicaciones, dado que toda comunicación que utilice internet como canal de comunicación, necesita exclusivamente de IP.

Estos avances tecnológicos han traído como beneficio, la creación del IPv6, protocolo de internet que no solo trae muchas ventajas en la seguridad de las redes, sino que soluciona el inconveniente del IPv4, frente al agotamiento de direcciones IP.

Lamentablemente, incluso IPv6 presenta puntos débiles tales como:

- IPv6 todavía re-encapsula por IPv4, por lo que el tráfico de IPv6 puede viajar por las redes sin que los administradores lo sepan [IPv00].
- La escasez de las herramientas de seguridad para estos protocolos nos lleva a hacer uso de los firewalls actuales. Sin embargo, muchos de éstos no soportan IPv6 [IPv00].
- La configuración automática de direcciones IPv6, permite que los hackers hagan uso de estas direcciones. De forma adicional, dado que IPv6 puede usar varias direcciones en una misma interfaz, esto también puede traer consigo problemas [IPv00].

- Siendo el tráfico de IPv6 mucho mayor, debido a que su cabecera es más grande, se produce latencia [IPv00].

Sin embargo, se debe tender a utilizar cada vez mas IPv6, debido a que sus ventajas sobre IPv4, lo compensan. Y siendo IPv6 la tecnología con la que actualmente se está trabajando, es necesario que se haga de IPv6 el protocolo de internet más seguro existente, y por ello debemos solucionar sus puntos débiles y mejorar esta tecnología.

Teniendo en cuenta las limitaciones del IPv6 y la existencia de los distintos tipos ataques en IPv6, se propone una mejora en la forma de autenticación ya existente para IPv6. Dado que el usuario debe tener la seguridad de que cualquier operación que realice sea lo más segura y transparente posible.

Finalmente conocer el IPv6 y trabajar sobre su autenticación es una tarea que se debe realizar, debido a que los estudios y conclusiones de muchos trabajos sobre la seguridad en las redes como [TEC14, Cal, Yan14, Gar12a, AD00, LB08], demuestran que el estudio de los puntos débiles de IPv6 en su autenticación es la base para corregir los errores en este protocolo.

1.2. Planteamiento del problema

La transición de IPv4 a IPv6 ha traído nuevos retos en la seguridad de las redes, pese a los diferentes aspectos que se han solucionado con IPv6, todavía no se puede asegurar una comunicación del todo segura entre puntos de comunicación. Cuando una comunicación se realiza a través de IPv6, esta pasa por un proceso de encriptación y autenticación. La autenticación adiciona campos propios a la información que se envía, los cuales son encapsulados, para que de esta manera el remitente sea el único que pueda descifrar la información.

El problema radica cuando este proceso de autenticación es burlado por otros usuarios, produciéndose una invasión en la privacidad de la información enviada o incluso una alteración en la información misma. Los estudios revelan que los ataques típicos en IPv6 e IPv4 son los siguientes [Alo12, Var13]:

- Descubrimiento de equipos.
- NDP Redirect.
- Hombre en el medio (MITM) sobre redes IPv4 utilizando ARP Spoofing.
- Hombre en el medio (MITM) sobre redes IPv4 utilizando inyecciones DHCP ACK.

- Hombre en el medio (MITM) sobre redes IPv6 utilizando Neighbor Advertisement Spoofing.
- Hombre en el medio (MITM) sobre redes IPv6 utilizando SLAAC Attack.
- Hombre en el medio (MITM) sobre redes IPv6 utilizando falso DHCPv6.
- DoS (Denial of Service) sobre redes IPv4 utilizando ARP Spoofing.
- DoS sobre redes IPv6 utilizando SLAAC Attack.
- DNS Hijacking.

Estos problemas nos muestran la necesidad de mejorar la autenticación en IPv6, y con ello garantizar una mayor confiabilidad y privacidad en las comunicaciones de los usuarios de internet.

1.3. Objetivos

1.3.1. Objetivo general

El objetivo general de este proyecto es desarrollar una mejora en la autenticación del protocolo IPv6, el cual incorporará técnicas del protocolo Kerberos. Mejora que busca solucionar el ataque de "el hombre en el medio".

1.3.2. Objetivos específicos

Para lograr el objetivo general de este proyecto, se tiene los siguientes objetivos específicos:

1. Analizar artículos de investigación con diferentes propuestas de autenticación IPv6, para determinar sus alcances, ventajas y desventajas de cada uno.
2. Proponer e implementar un nuevo protocolo de autenticación en IPv6, incorporando técnicas del protocolo Kerberos, el cual solucione el ataque del MITM.
3. Realizar las pruebas de evaluación respectivas, analizar los resultados obtenidos, comparar los alcances de la nueva propuesta con la autenticación IPv6 ya analizada y presentar las conclusiones finales.

1.4. Metodología

La metodología para este proyecto, está desarrollada de la siguiente manera:

1. Investigar y estudiar otros trabajos previos: Se busca investigar, recopilar, analizar y estudiar la forma en que IPv6 realiza su autenticación, con el fin de poder conocer sus alcances y limitaciones. Este conocimiento conlleva estudiar muchos otros conceptos, además de la autenticación, con el fin de proponer un nuevo prototipo de autenticación en IPv6, que solucione una de las tantas limitaciones que se encontrarán. Este punto conlleva también estudiar otros trabajos sobre seguridad en IPv6, para conocer sus alcances actuales. Entre los conceptos que se deberán profundizar, se tienen los siguientes:
   - El protocolo IPv4 e IPv6.
   - El protocolo IPsec (IP Security).
   - La cabecera de autenticación de IPsec.
   - La cabecera de encapsulado de la carga de seguridad de IPsec.
   - El protocolo IKE (Internet Key Exchange).
   - El protocolo Kerberos.
   - La relación existente entre Kerberos e IKE.
   - La relación existente entre Kerberos e IPv6.
   - El protocolo PGP (Pretty Good Privacy).
   - El Iprotocolo ISAKMP (Internet Security Association and Key Management Protocol).
2. Proponer un novedoso prototipo del protocolo de autenticación en IPv6: Habiendo estudiado todos los conceptos necesarios, se llegaron a dos conclusiones. La primera es que la base para mejorar la autenticación es trabajar sobre IPsec, y la segunda que las aplicaciones que envían una contraseña no cifrada por la red son vulnerables. Estas ideas, acompañadas con otros estudios, nos llevaron a optar por el uso del protocolo Kerberos. Debido a que Kerberos no trabaja con contraseñas, sino con la base de tickets para demostrar la identidad de los usuarios, además de que usa claves secretas para realizar una comunicación. Se busca incorporar Kerberos a IPv6, tomando las ventajas de Kerberos y adicionándolas a IPv6 para crear un nuevo protocolo que sea inmune al ataque de MITM. De esta forma se creó un nuevo protocolo de autenticación, el cual operará como la forma tradicional de IPv6, con la diferencia que autenticará como Kerberos, sin escapar de los procedimientos para la autenticación entre pares, gestión de asociaciones de seguridad y mitigación amenazas que son definidos por el ISAKMP.

3. Implementar la nueva propuesta de autenticación: Una vez que se haya propuesto el nuevo protocolo y se haya determinado su operatividad, se procede a utilizar un software de simulación de redes o un lenguaje de programación

que nos permita implementarlo. En este caso se optó por implementar la nueva propuesta en C y realizar las simulaciones respectivas con Linux, a partir del uso del terminal. El lenguaje de programación C para Linux, nos permite implementar una comunicación entre dos o más terminales de Linux. El primer paso es enviar un paquete IP, desde un terminal de Linux con un mensaje, hacia otro terminal de Linux. Ambos terminales deberán estar en modo root. Una vez que se haya enviado un paquete, este paquete deberá ser recibido por el segundo terminal y se deberá poder acceder a su información, para determinar la veracidad y autenticidad del emisor. Habiéndose creado el transmisor y receptor de mensajes IP, y permitiendo una comunicación entre 2 nodos, se procede a realizar la comunicación, sin embargo esta vez se realizará entre tres nodos. Este tercer elemento se crea para permitir la autenticación con Kerberos. Las pruebas necesarias se podrán realizar mediante diferentes comandos desde los terminales. Finalmente se procede a implementar la nueva propuesta denominada como "IPv6K". Lo que se busca con esta nueva propuesta es hacer uso del protocolo Kerberos, el cual no utiliza passwords para garantizar una mejora en la autenticación de IPv6.

4. Realizar las pruebas respectivas: En esta etapa se realizará la comparación respectiva de la nueva propuesta de autenticación implementada, con la forma de autenticación ya existente de IPv6. Para ello se crearán diferentes casos de pruebas, y se enviarán una serie de paquetes IPv6, utilizando las dos formas de autenticación, en el entorno de simulación. Este punto comprende el análisis y toma de resultados de las comparaciones que se realizarán, así como la obtención de los resultados de las pruebas, las cuales servirán para la última etapa.

5. Conclusiones de los datos obtenidos: Mediante las respuestas obtenidas de la comparación de autenticaciones se presentarán las conclusiones finales con los resultados de dichas pruebas. Se dará a conocer los alcances de la nueva propuesta y se presentarán sus puntos a mejorar para trabajos futuros.

1.5. Consideraciones complementarias

1.5.1. Recursos y materiales

La lista de recursos y materiales necesarios para este proyecto, comprende ciertos equipos con los que cuenta la Universidad Católica San Pablo, como único recurso adicional, se necesita de un software de simulación para la propuesta, a continuación mencionaremos los recursos y materiales necesarios:

- Una computadora de escritorio o una computadora portátil: Para poder realizar las simulaciones de la nueva propuesta, siendo el ordenador nuestro medio de trabajo para simular y probar la nueva propuesta.
- Plataforma de Ubuntu o máquina virtual con Ubuntu: Como hemos mencionado anteriormente, se busca implementar, programar y simular en C en la plataforma de Linux. Este software solo puede ejecutarse desde la plataforma de Ubuntu. De contar con un ordenador de 64 bits y hacer uso de una máquina virtual, se necesitará de una plataforma de Ubuntu para 64 bits. En este caso se ha necesitado de Ubuntu 14.04.2.

1.5.2. Cronograma de trabajo

Este proyecto presenta una duración de 10 meses. El proyecto inició desde el momento en que se comenzó a buscar información de este tema, con el fin de contar con un vasto conocimiento de todos los temas que implican en este proyecto, su conclusión es la presentación final, mediante un informe detallado. De forma detallada los pasos seguidos hasta su finalización han sido los siguientes, mostradas en las siguientes Figuras 1.1 y 1.2:

DIAGRAMA DE GANTT

	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1		Proyecto de Tesis	210 días	lun 17/08/15	vie 03/06/16
2		Investigar Información sobre IPv4 e IPv6.	28 días	lun 17/08/15	mié 23/09/15
3		Estudiar el funcionamiento de la Autenticación de IPv6.	14 días	jue 24/09/15	mar 13/10/15
4		Investigar los Ataques a la Autenticación de IPv6.	14 días	jue 01/10/15	mar 20/10/15
5		Investigar Información sobre Ipsec, Hash, IKE y Kerberos.	14 días	jue 08/10/15	mar 27/10/15
6		Analizar los Alcances Desarrollados Sobre Mejoras en IPv6.	14 días	jue 15/10/15	mar 03/11/15
7		Proponer un Prototipo de Mejora en la Autenticación de IPv6 que no haya sido Desarrollado.	14 días	vie 16/10/15	mié 04/11/15
8		Analizar y Conseguir el Software de Simulación que Ayudara con el Diseño del Prototipo.	14 días	vie 23/10/15	mié 11/11/15
9		Diseñar el Funcionamiento del Prototipo, que se planea desarrollar con el Software de Simulaciñon.	14 días	vie 30/10/15	mié 18/11/15
10		Realizar la Primera Revisión del Diseño Inicial del Prototipo.	7 días	jue 19/11/15	vie 27/11/15
11		Realizar la Segunda Revisión del Diseño Inicial del Prototipo.	7 días	sáb 28/11/15	lun 07/12/15
12		Redactar al Programa Lyx, toda la Informaciòn Recopilada siguiendo el Formato.	52 días	jue 01/10/15	vie 11/12/15
13		Implementar la Simulación de las Unidades de la Propuesta.	40 días	vie 18/12/15	jue 11/02/16
14		Realizar Pruebas de Evaluación de las Unidades de la Propuesta.	20 días	vie 12/02/16	jue 10/03/16
15		Integrar las Unidades Implementadas.	25 días	vie 11/03/16	jue 14/04/16
16		Realizar Pruebas de Evaluación de las todas las Unidades de forma Integrada.	20 días	vie 15/04/16	jue 12/05/16
17		Evaluar los Resultados de la Nueva Propuesta Integrada.	10 días	vie 13/05/16	jue 26/05/16
18		Realizar el Analisis Comparativo entre la Nueva Propuesta y el Metodo Tradicional de Autenticación.	5 días	vie 27/05/16	jue 02/06/16
19		Mostrar las Conclusiones Finales.	5 días	jue 02/06/16	mié 08/06/16
20		Redacción Final del Informe Favorable.	124 días	vie 18/12/15	mié 08/06/16

Figura 1.1: Actividades a realizar del cronograma de trabajo.

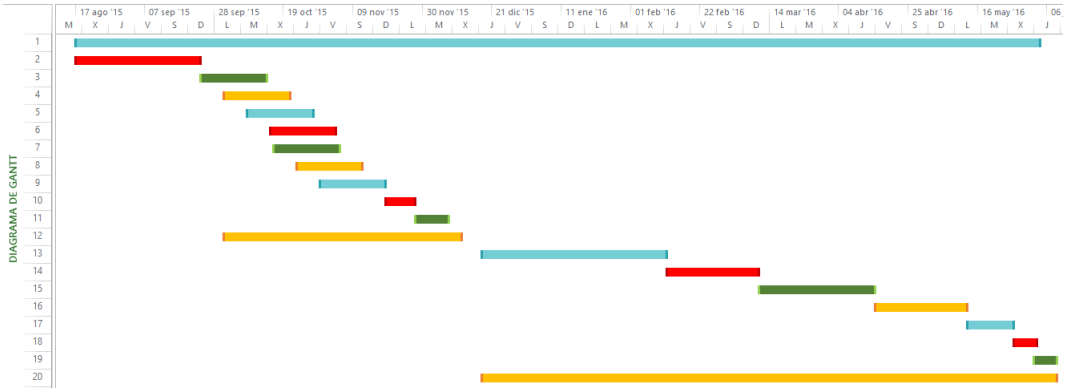

Figura 1.2: Cronograma de trabajo para el presente proyecto.

Organización del presente documento

El presente trabajo se encuentra organizado de la siguiente manera:

- Se inicia con el segundo capítulo, el cual presenta el marco teórico del proyecto. En este capítulo se detallan todos los conceptos teóricos que se deben conocer, para tener las bases necesarias del desarrollo planteado. Los puntos que se desarrollaran en el marco teórico son los siguientes:
  - ¿Qué el protocolo IPv6? Aquí se desarrollan los conceptos más importantes de este protocolo, pero principalmente se detalla el proceso de autenticación en IPv6.
  - La teoría del protocolo IPsec. En este punto se presentan los campos que conforman al IPsec, que beneficios ofrece este protocolo, y principalmente se introducen los conceptos de cabecera de Autenticación en IPsec y en Hash.
  - La cabecera IKE. A partir de los conceptos que nos presenta la cabecera IKE nos adentramos aún más en la autenticación, ya que en el IKE se encuentra el corazón de la autenticación.
  - El protocolo Kerberos. Se desarrolla este protocolo dado su gran utilidad en la autenticación y compatibilidad con IPv6.
  - Unión de IKE y Kerberos. En este punto se muestra la compatibilidad de estos conceptos.
  - Los ataques a las redes IPv6. Se presentan de forma detallada todos los ataques en IPv6 y el impacto del MITM.

  - El Internet Security Association y Key Management Protocol. Finalmente se da a conocer los diferentes tipos de cargas útiles que se envían en una comunicación.

- En el tercer capítulo se presenta el estado del arte. Se desarrolla la discusión de los aspectos fuertes y alcances actuales de los últimos trabajos y proyectos que comprenden el estudio del IPv6. Lo que se busca en este capítulo es conocer hasta qué punto ya se ha desarrollado el tema a investigar en otros trabajos, para de esta forma tener la base de cómo atacar el problema que deseamos solucionar, y cómo realizar las mejoras respectivas.
- En el cuarto capítulo se presenta la propuesta de solución a la problemática a desarrollar. Se desarrolla y explica de forma detallada como la propuesta IPv6K operará, presentandose distintos escenarios de comunicación, en los cuales se ve cómo la propuesta IPv6K opera para estos escenarios, mostrando la operatividad de la propuesta y su escala para diferentes escenarios de comunicación.
- En el quinto capítulo se presenta el desarrollo, implementación y simulación de la propuesta de solución. A partir de las explicaciones mostradas en el cuarto capítulo se desarrolla la implementación de la propuesta IPv6K, se realizan las simulaciones en un escenario de comunicación y se muestran los resultados obtenidos.
- En el sexto y último capítulo se presenta las conclusiones finales referentes al trabajo de investigación en general, los alcances de la propuesta y los trabajos que se pueden realizar a futuro.

2 Marco teórico

2.1. IPv6

Tal y como nos lo señala el estándar RFC2460 [Rfc], IPv6 es un estándar de los protocolos de internet, desarrollado por el grupo de trabajo de ingeniería de internet (IETF), con el objetivo de ser el sucesor del IPv4.

Como nos lo indica García en [Gar12b], IPv4 fue desarrollado por el estándar RFC791 [Pos81], con el objetivo de ser el primer protocolo de internet que garantizaría una comunicación de datos entre usuarios. De acuerdo con [Pos81, Gar12b], se planeó que la capacidad de direcciones de IPv4 fuera suficiente para cubrir a todos sus usuarios durante varios años, sin embargo no se anticipó que el número de usuarios crecería en gran medida en pocos años.

Es por esta razón, que fue necesario el desarrollo de un nuevo protocolo de internet, que no solo fuera capaz de cubrir el gran número de usuarios, sino que también solucione los problemas que IPv4 no había podido solucionar, problemas con seguridad, privacidad de los datos, priorización de los datos, etc.

El nuevo protocolo de internet, desarrollado por el estándar RFC1883 [DH95], se denominó IPv6 y con su aparición se mejoraron muchos de los aspectos de seguridad del IPv4, también se eliminaron campos que resultaban redundantes. Entre los beneficios que trajo IPv6, y basándonos en [Rfc, Pos81, Gar12a] podemos mencionar:

- Mayor expansión en capacidad de direcciones IP, aumentando a 128 bits.
- Mejoramiento de la escalabilidad de enrutamiento multicast.
- Inclusión de la dirección “anycast", la cual se utiliza para enviar un paquete a cualquiera en un grupo de nodos.
- Simplificación de la cabecera, eliminando campos redundantes.
- Mejoramiento del soporte para las extensiones y opciones, lográndose un reenvío más eficiente y mayor flexibilidad para introducir opciones.
- Capacidad de flujo de etiquetado, permitiéndose el etiquetado de paquetes que soliciten un tratamiento especial.
- Cabecera de extensión, para el mejoramiento del manejo de opciones.

- Autoconfiguración de direcciones libre de estado o SLAAC, gracias a este protocolo los nodos pueden determinar sus propias direcciones IPv6.
- Uso de jumbograms, logrando el envío de paquetes con cargas mayores.
- Mayor seguridad, facilidad de cifrado y autenticación de comunicaciones.
- Mayor movilidad al contar con un manejo más simplificado de los nodos.

2.2. IPsec

Para empezar el IPsec, tal y como nos lo explica el compendio de IPsec presentado en [dSODFU], es un conjunto de protocolos de seguridad de internet exclusivos del IPv6, cuya función es la de asegurar que se realicen las comunicaciones basadas con IP.

Este proceso es realizado mediante el autenticando y/o cifrando de cada paquete IP en su flujo de datos. A diferencia de otros protocolos, IPsec no necesita modificar su código cuando se realiza un cambio específico, además de actuar sobre la capa de transporte, también actúa sobre la capa de red haciendo que IPsec sea más flexible.

De forma adicional como nos lo señala [Pro05], IPsec actúa como un armazón de protocolos de seguridad y proporciona integridad de datos, autenticación y confidencialidad, además de asociaciones de seguridad (SA) y administración de claves. Entre los servicios que IPsec ofrece podemos mencionar los siguientes:

- Confidencialidad: La información únicamente es accesible por las entidades autorizadas.
- Integridad: Mediante los códigos detectores de errores IPsec garantiza que la información no será modificada.
- Autenticación: IPsec corrobora que el usuario es realmente quien dice ser.
- Acceso remoto a ordenadores en distintos lugares: Las redes privadas pueden comunicarse con diferentes sedes sin la necesidad de redes físicas privadas.
- Negociación del cifrado: IPsec permite al host emisor y receptor acordar las claves y algoritmos que negocien.
- Cifrado de la comunicación: Cuando el host emisor recibe un segmento, IPsec cifra la carga útil haciendo que estos datos sólo puedan ser descifrados por el host receptor.

2.2.1. Los protocolos de IPsec

Como nos lo presenta Castro en [CB10], el IPsec consta de tres protocolos, dos son los principales y han sido desarrollados con el objetico de proporcionar seguridad a

nivel de paquete, tanto para IPv4 como para IPv6. Los dos protocolos principales son:

- La cabecera de autenticación (AH).
- La cabecera de encapsulado de la carga de seguridad (ESP) .

Existe un tercer protocolo de IPsec, el cual es el siguiente:

- El internet key exchange (IKE)

2.2.1.1. La cabecera de autenticación

De acuerdo con los conceptos planteados en [dSODFU], la cabecera de autenticación está dirigida a garantizar la integridad sin conexión, y al mismo tiempo a garantizar la autenticación de los datos de origen en los datagramas IP. Esta tarea se realiza calculando un Hash Message Authentication Code (HMAC), a través de un algoritmo hash el cual opera tanto sobre una clave secreta, como sobre el contenido del paquete IP y las partes inmutables del datagrama.

De forma adicional, el AH brinda protección contra ataques de repetición (anti-replay), utilizando la técnica de ventana deslizante y descartando paquetes viejos. Finalmente, el AH protege la carga útil IP, y todos los campos de la cabecera de un datagrama IP, con excepción de aquellos campos que pueden ser alterados en el camino.

HASH

Como nos lo presenta el libro [Pro05], los hash son algoritmos especiales, los cuales a partir de una entrada consiguen crear una cadena de dígitos (salida alfanumérica) llamada “hash”, la cual es única para el contenido del mensaje.

El hash tambien cumple las funciones de asegurar que no se haya modificado un archivo en una transmisión, hacer ilegible una contraseña, sin embargo un hash no es regresivo, es decir que un mensaje puede producir un hash, pero un hash no puede producir el mensaje original.

HMAC

El código de autenticación de mensajes basado en hash (HMAC), es un tipo de mecanismo de gran utilidad para IPsec que como nos lo presentan [Tec, Wol, KB, Pro05] permite garantizar la integridad del mensaje. En el origen el mensaje se combina con una clave secreta compartida, estos son enviados a través de un algoritmo de hash.

Y como nos mencionan [Tec, Pro05], serán los códigos de autenticación de mensajes (MAC), lo que proporcionarán la forma de comprobar la integridad de la información transmitida. Para transmitir usando HMAC, el mensaje y el hash se envían a través de la red habitualmente enganchados el uno al otro. Mientras que en el lado del receptor, se lleva a cabo el siguiente proceso:

1. El mensaje recibido y la clave secreta compartida se envían a un algoritmo hash para recalcular el valor hash.
2. El receptor comparará ambos valores de hash, en el caso de que coincidan la integridad del mensaje está garantizada.

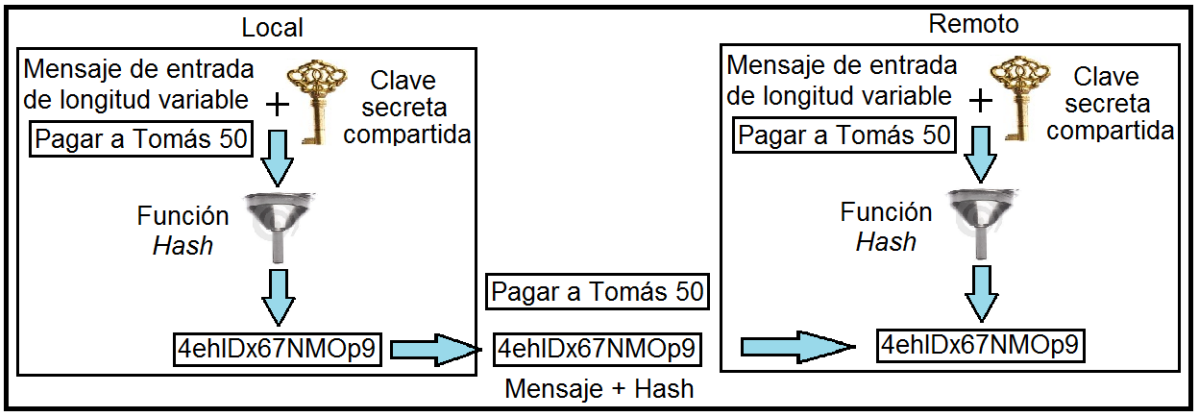

Figura 2.1: HMAC. [Pro05]

2.3. Internet key exchange

Yang nos indica en [Yan14], que el IKE es un protocolo estandarizado de IPsec, el cual es utilizado para garantizar la seguridad en negociaciones de las VPNs y en los host remotos o en el acceso a la red. El protocolo IKE mejora el IPsec, al proporcionar características adicionales y mayor flexibilidad, además de hacer que IPsec sea mucho más sencillo de configurar.

Como nos lo presenta el libro [Pro05], el protocolo IKE ofrece autenticación en IPsec, además de ofrecer negociado de claves y negociado de las asociaciones de seguridad del IPsec. De forma adicional el protocolo IKE garantiza la seguridad para las comunicaciones de SAs, sin la pre-configuración de otro modo que fuera necesario.

La configuración del modo IKE permite que un gateway descargue en el cliente una dirección IP y otro tipo de parámetros a nivel de red, como parte de la negociación IKE [Pro05]. A través de este intercambio, el gateway entrega direcciones IP al

cliente IKE para que sean utilizadas como una dirección IP interna encapsulada bajo IPsec. Este proceso ofrece una dirección IP conocida al cliente, la cual puede obtenerse mediante una política IPsec [Pro05]. Podemos observar la forma en que IPsec se utiliza en la figura 2.2.

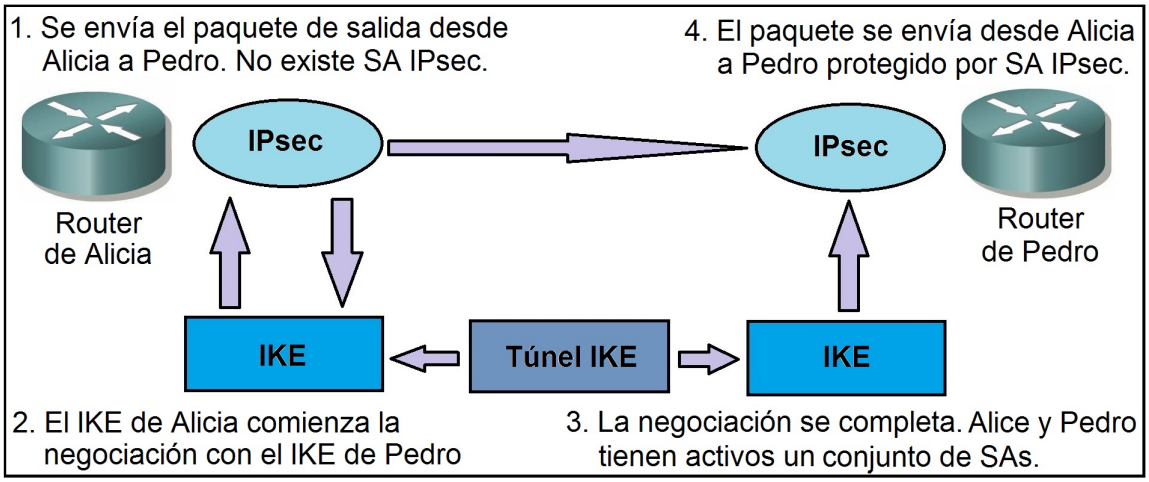

Figura 2.2: ¿Cómo se usa IPsec en el IKE? [Pro05]

La situación actual sobre el protocolo de internet key exchange, es que no permite una autenticación segura entre peers, cuando se utilizan contraseñas. Es por ello que existen varias propuestas para integrar autenticación por contraseña en los protocolos de IKE.

2.4. Protocolo Kerberos

Es evidente la transferencia de contraseñas sin encriptar sobre la red, bajo los protocolos FTP y Telnet. Dado que la autenticación de los usuarios a los servicios de red, se vuelve insegura cuando el método utilizado por el protocolo también es inseguro, es necesario un método que garantice la autenticación a nivel de IP, y es por ello que se necesita del protocolo Kerberos, el cual es un protocolo que permite a dos ordenadores en una misma red insegura, demostrar su identidad mediante el uso de tickets.

Tal y como nos lo presenta el libro [Hat05] Kerberos es un protocolo que utiliza la criptografía de claves simétricas, para validar a los usuarios con los servicios de red. Kerberos requiere de un tercero de confianza y se basa en el protocolo de Needham-Schroeder. Este tercero de confianza, denominado el "centro de distribución de claves" (KDC), consiste en un "servidor de autenticación" (AS) y un "servidor emisor

de tickets" (TGS).

Kerberos mantiene una base de datos de claves secretas, es decir que cada entidad en la red, comparte una clave secreta conocida únicamente por la red y Kerberos [Hat05].

2.4.1. Ventaja principal del Kerberos

Como ventaja sobre el protocolo Kerberos, indicaremos primero que los servicios de redes más convencionales usan esquemas de autenticación basados en contraseñas, es decir que tales esquemas requieren que cuando un usuario necesite de una autenticación en un servidor de red, el servidor debe proporcionar un nombre de usuario y una contraseña. Sin embargo la información de autenticación para muchos servicios se transmite sin estar encriptada [Hat05].

Una vez que la red se conecte a la internet, ya no se puede asumir que la red es segura. Cualquier intruso del sistema con acceso a la red y un analizador de paquetes pueden interceptar cualquier contraseña enviada de este modo, comprometiendo las cuentas de usuarios y la integridad de toda la infraestructura de seguridad [Hat05].

Kerberos busca eliminar la transmisión a través de la red de información de autenticación. Es decir que su principal objetivo es el de prevenir que las contraseñas no encriptadas sean enviadas a través de la red. Por lo que erradica la amenaza de analizadores de paquetes que intercepten contraseñas en su red [Hat05].

2.4.2. Desventaja principal del Kerberos

A pesar de que Kerberos elimina una amenaza de seguridad común, puede ser difícil de implementar por las siguientes razones [Hat05]:

- La migración de contraseñas de usuarios desde una base de datos de contraseñas estándar UNIX, a una base de datos de contraseñas Kerberos puede ser tediosa y no hay un mecanismo rápido para realizar esta tarea.
- Kerberos es sólo parcialmente compatible con los Pluggable Authentication Modules (PAM) usados por la mayoría de los servidores Red Hat Enterprise Linux.
- Kerberos presupone que cada usuario es de confianza sin embargo puede darse el caso de que ésta se esté utilizando una máquina no fiable en una red no fiable.

- Para que una aplicación use Kerberos, el código debe ser modificado para hacer las llamadas apropiadas a las librerías de Kerberos. Las aplicaciones que son modificadas de esta forma son consideradas kerberizadas.
- Si se decide usar Kerberos en una red, debe darse cuenta de que es una elección de todo o nada. Si decide usar Kerberos en su red, debe recordar que si se transmite cualquier contraseña a un
- Finalmente, para asegurar su red con Kerberos, debe utilizar las versiones kerberizadas (que funcionen con Kerberos) de todas las aplicaciones cliente/servidor que envíen contraseñas sin encriptar o no utilizar ninguna de estas aplicaciones en la red.

2.4.3. Terminología de Kerberos

A. El Centro de distribución de llaves (KDC)

Es un servicio que emite tickets Kerberos, que habitualmente se ejecutan en el mismo host que el servidor de otorgamiento de tickets (TGS) [Hat05].

B. El servidor de otorgamiento de tickets (TGS)

Es un servidor que emite tickets para un servicio deseado; estos tickets son entregados a los usuarios para que acceda el servicio. El TGS usualmente se ejecuta en el mismo servidor que KDC [Hat05].

C. El Servidor de autenticación (AS)

Es un servidor que emite tickets para un servicio deseado los cuales a su vez son entregados a los usuarios para que accedan al servicio. El AS responde a las peticiones de los clientes quienes no tienen o no envían credenciales con la petición. El AS usualmente se ejecuta en la misma máquina que el KDC [Hat05].

D. Las credenciales

Son un grupo temporal de credenciales electrónicas que verifica la identidad de un cliente para un servicio particular. También se conoce como ticket [Hat05].

E. El caché credencial o archivo de tickets

Es un archivo que contiene las llaves para encriptar las comunicaciones entre el usuario y varios servicios de red [Hat05].

F. El hash encriptado

Es un hash de un sentido usado para autentificar usuarios. Aunque es más seguro que los datos sin encriptar [Hat05].

G. El GSS-API

La interfaz del programa de la aplicación de servicio de seguridad genérico, mejor conocido como GSS-API (Generic Security Service Application Program Interface), es un conjunto de funciones que proveen servicios de seguridad. Esta API es usada por clientes y servicios para autenticarse entre ellos sin que ninguno de los programas tenga un conocimiento específico del mecanismo detrás de ello. Si un servicio de red usa GSS-API, puede autenticar usando Kerberos [Hat05].

H. Las llaves o keys

Son datos usados cuando encriptamos o des-encriptamos otros datos. Los datos encriptados no pueden ser desencriptados sin la llave apropiada o con una extraordinaria capacidad para adivinar [Hat05].

I. El principal (o nombre del principal)

El principal es el nombre único de un usuario o servicio que puede autenticar mediante el uso de Kerberos [Hat05].

J. El ticket

Es un grupo temporal de credenciales electrónicas que verifican la identidad de un cliente para un servicio particular. También llamado credenciales [Hat05].

K. El ticket granting ticket (TGT)

Es un ticket especial usado por los clientes, con el propósito de acceder a otros servicios de Kerberos [Hat05].

2.4.4. Modo en que funciona Kerberos

Tal y como nos lo explican [McC13, Pea14], empezaremos diciendo que el funcionamiento de Kerberos, se realiza con 3 componentes claves, estos 3 componentes estarán presentes durante toda negociación y proceso de autenticación. Estos 3 componentes son:

- El cliente o usuario de Kerberos o simplemente el usuario.

- El servidor de Kerberos o simplemente el servidor.
- El centro de distribución de llaves (KDC).

Ahora cabe mencionar que Kerberos es diferente de los métodos de autenticación de usuario/contraseña, esto se debe a que el protocolo Kerberos permite la autenticación sin la necesidad de transmitir passwords a través de la red. Y como nos lo presentan [McC13, Pea14], de forma adicional Kerberos permite a los usuarios acceder a múltiples servicios. El proceso de autenticación y negociación que realiza Kerberos, es el siguiente:

1. Como se observa en la Figura 2.3, primero se realiza la autenticación del cliente Kerberos al servidor KDC, este proceso de autenticación se realiza mediante un proceso simple de logueo de este cliente. Al autenticarse el cliente de Kerberos al KDC, el cliente envía un paquete con un mensaje de identificación al KDC. Este mensaje solo contiene el nombre del cliente y la petición del cliente de solicitar al servidor KDC un TGT. Todo TGT que sea generado por el KDC, tiene una línea o periodo de tiempo específico, como parte de su seguridad, una vez agotado dicho tiempo el ticket será destruido [McC13, Pea14].

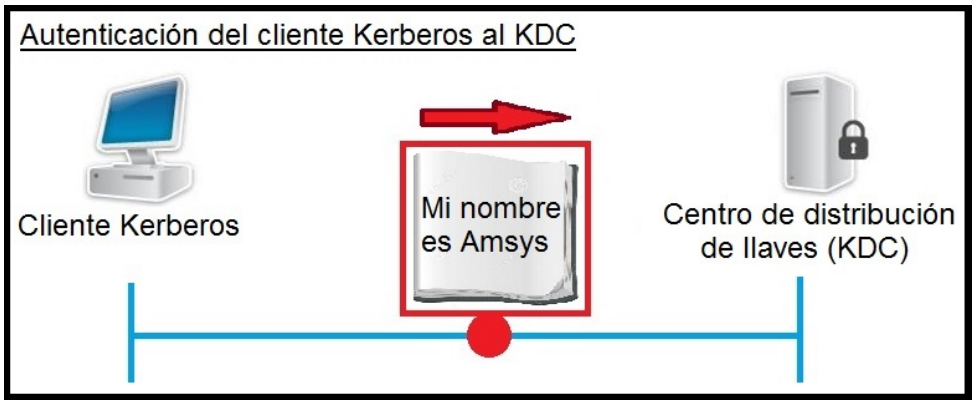

Figura 2.3: Envió del nombre del cliente al KDC.

2. Como se aprecia en la Figura 2.4, la petición y la identidad del cliente Kerberos serán verificadas por el KDC, quien cuenta con una base de datos de passwords. Si el servidor KDC corrobora la identidad del cliente en su base de datos, entonces el servidor KDC producirá una TGT para el cliente. Cabe mencionar que el TGT contiene la siguiente información:

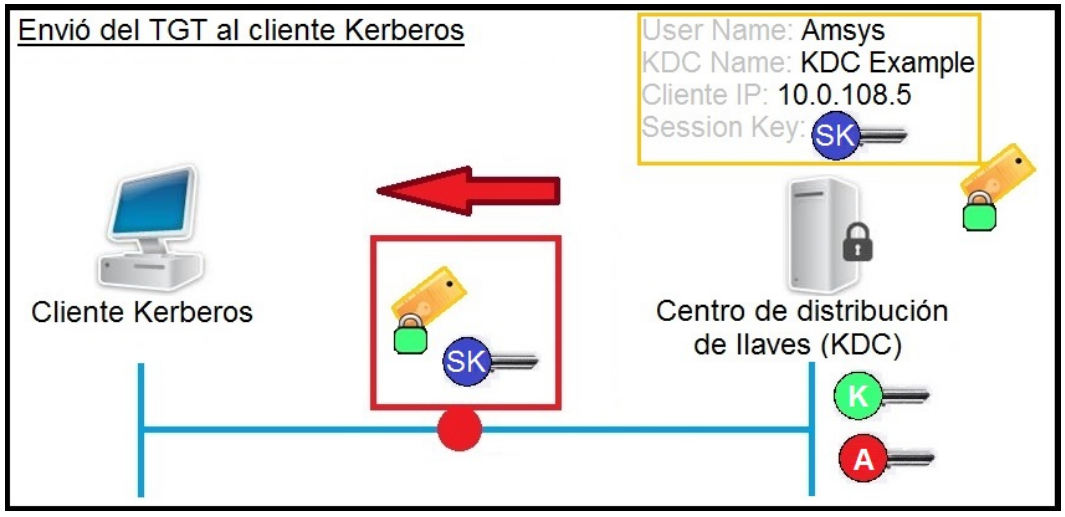

Figura 2.4: Envió del TGT al cliente.

- El nombre del cliente
- El nombre del KDC
- La IP del cliente
- La llave de sesión

Esta llave de sesión (Session Key) es una llave de encriptación con línea de tiempo, creada exclusivamente para la presente sesión, la cual cambia progresivamente, como medida de seguridad.

El servidor KDC encriptará la TGT generada, utilizando su propia llave privada, solo el servidor KDC tiene dicha llave, por lo que el cliente no podrá ver el contenido del TGT. Una vez encriptado el TGT el servidor KDC crea un paquete donde enviara el TGT encriptado, junto con la llave de sesión del TGT. Dicho mensaje se encriptara con la llave privada del cliente y se enviara al cliente [McC13, Pea14].

3. Continuando y como se observa en la Figura 2.5, el usuario o cliente Kerberos, des-encriptará el paquete recibido con su llave privada y recibe tanto el TGT como la llave de sesión. En ese momento el cliente crea un “Autenticador”, el cual contiene:

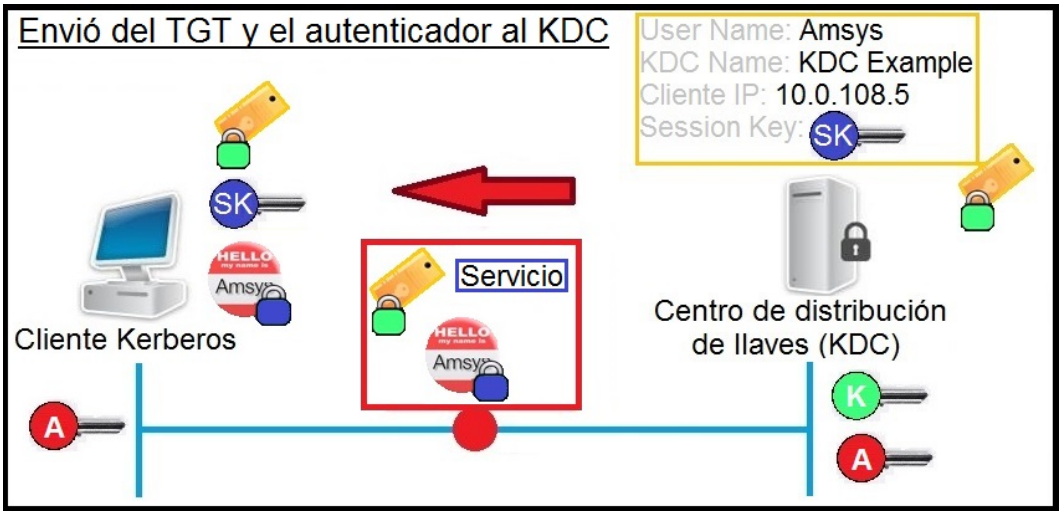

Figura 2.5: Envió del TGT y del autenticador al KDC.

- El nombre del cliente.
- La IP del cliente.
- El tiempo.

Este autenticador será encriptado por el cliente con la llave de sesión que recibió y se enviará un paquete nuevamente al KDC con "el autenticador encriptado", con el "TGT" y con el nombre del "servicio que se desea solicitar" [McC13, Pea14].

4. Como se presenta en la Figura 2.6, el servidor KDC recibirá el mensaje y desencripta el TGT utilizando su llave privada, una vez corroborado que el TGT no ha sido abierto ni alterado, el KDC verifica la identidad del cliente con el autenticador que recibió. De ser positiva la identidad del cliente, el KDC crea un nuevo ticket llamado "ticket de servicio", el cual es exclusivo para el servicio que se solicita. El ticket de servicio también presentará una línea de tiempo y contendrá la siguiente información:

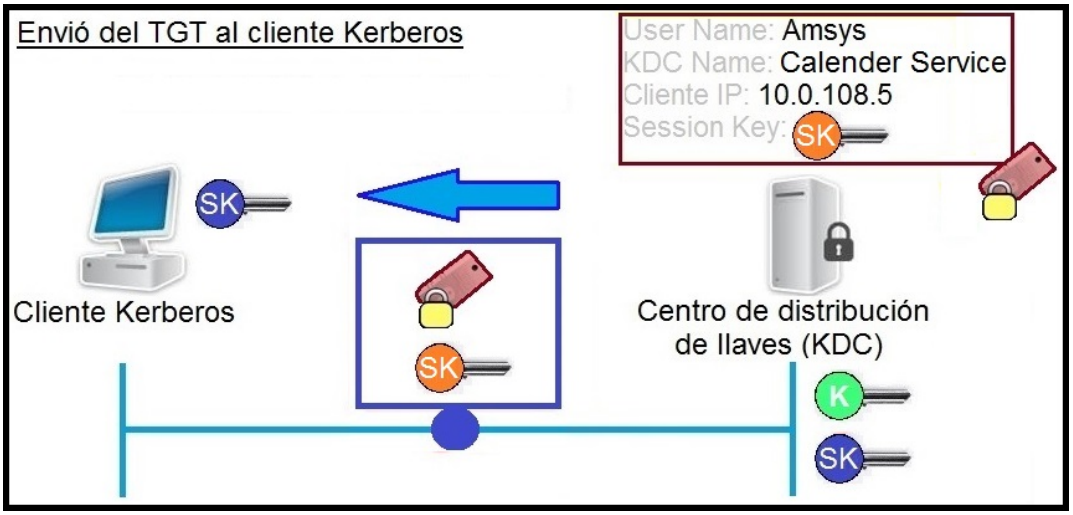

Figura 2.6: Envió del ticket de servicio y la nueva llave de sesión al cliente.

- El nombre del cliente.
- El servicio solicitado.
- La IP del cliente.
- Una nueva llave de sesión.

El servidor KDC encripta el ticket de servicio con una "llave privada de servicio". De forma similar al TGT, el cliente no tiene una copia de esta "llave privada de servicio", por lo que no podrá accesar a la información del ticket de servicio. El servidor KDC crea un mensaje cuyo contenido será el "ticket de servicio encriptado", y una copia de la "nueva llave de sesión, todo este mensaje se encriptará con la antigua llave de sesión y se enviará al cliente [McC13, Pea14].

5. A continuación y como se muestra en la Figura 2.7, el cliente des-encriptara el mensaje recibido con la antigua llave de sesión, recibiendo el ticket de servicio y la nueva llave, a partir de este momento el cliente deja de comunicarse con el KDC y procede a comunicarse con el servidor Kerberos. El cliente creara un nuevo autenticador y encriptara dicho autenticador con la nueva llave de sesión recibida del KDC, luego procede a enviar un paquete al servidor de Kerberos, con su "autenticador encriptado" y el "ticket de servicio" que recibió del KDC [McC13, Pea14].

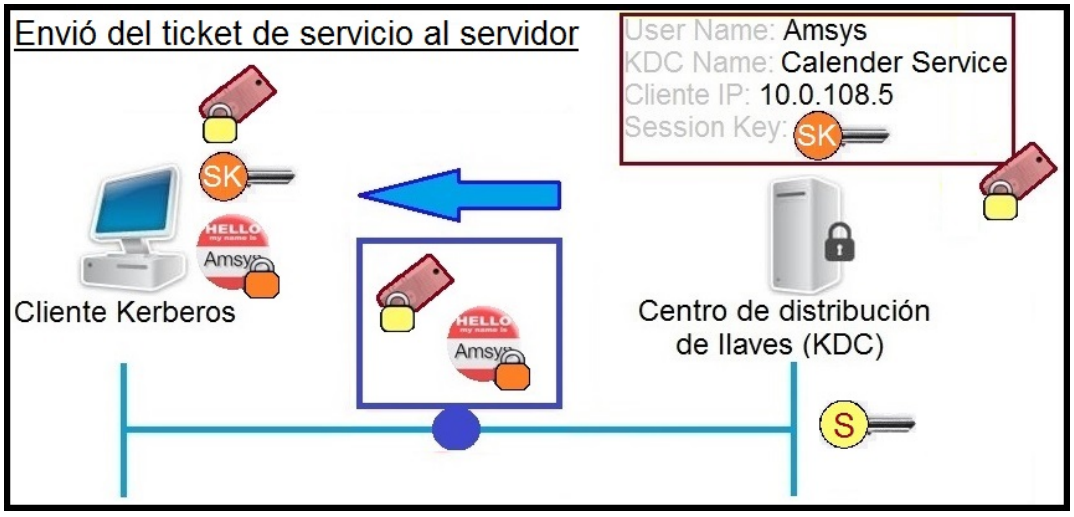

Figura 2.7: Envió del ticket de servicio y el autenticador al servidor de Kerberos.

6. Finalmente y como se observa en la Figura 2.8, el servidor de Kerberos utilizara su llave privada de servicio para des-encriptar el ticket de servicio. Sabiendo el servidor de Kerberos que la información del ticket fue generada del KDC, el servidor utiliza la nueva llave de sesión que se encuentra dentro del ticket de sesión, para acceder al autenticador que recibió. El servidor confirma la identidad del cliente utilizando el ticket de servicio y el autenticador, y de ser positiva la identidad del cliente, el servidor enviará el servicio solicitado al cliente y valida su uso [McC13, Pea14].

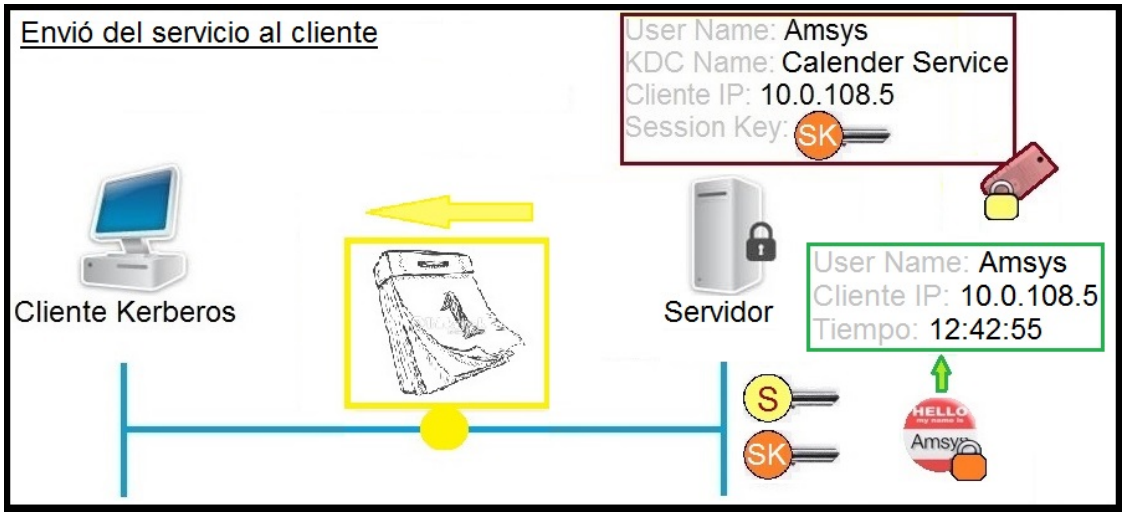

Figura 2.8: Envió del servicio solicitado al cliente.

2.5. IKE y Kerberos

2.5.1. IKE y la autenticación GSS-API

Primero como nos lo presenta el libro [LB08] una de las más sorprendentes fuentes de perdida de información es el protocolo Internet Key Exchange (IKE). IKE está diseñado para proteger la identidad de los participantes contra sniffing. Esta protección se logra, primero mediante la realización de un intercambio no autenticado de claves Diffie-Hellman y encriptando la siguiente autenticación con la clave de sesión. La protección de la identidad es considerada una de las principales características de seguridad de IKE.

Los métodos de autenticación estándar de IKE se basan en compartir llaves y en certificar llaves públicas. Windows extiende esta autenticación, con la autenticación Kerberos utilizando el GSS-API. El cliente solicita la autenticación Kerberos en el primer mensaje que envía al servidor. El cliente obtiene entonces un ticket de Kerberos del centro de autenticación (AC) y utiliza este ticket para la autenticación en IKE [LB08].

La mayor pérdida de información sucede porque el método de autenticación GSSAPI envía el nombre del equipo cliente y el dominio al servidor en el primer mensaje IKE (en la carga útil SA). Esto puede no parecer un problema de privacidad ya que la autenticación de Kerberos se usa sólo en la intranet y, por lo tanto, los datos nunca deben ser enviados cuando el ordenador está recorriendo en una red de acceso extranjera [LB08].

En realidad, la pérdida se produce, cuando el cliente se mueve de la intranet, a una red de acceso de extranjeros y las aplicaciones todavía intentan conectarse a servidores de intranet basado en direcciones IP anteriormente resueltas.

2.5.2. Solicitud de tickets Kerberos

Los clientes de Windows a veces intentan conectarse al servidor Kerberos, mientras este está corriendo. La solicitud de tickets contiene el nombre del equipo del cliente en texto plano. Dado que este es un caso raro, no se puede establecer la causa exacta de la solicitud. Debido a que el servidor de Kerberos normalmente no es accesible desde fuera de la intranet, el cliente no recibirá un ticket.

Si lo hiciera, el ticket revelaría aún más el nombre del servidor al cual está destinado el ticket [LB08]. Sin embargo y como presentaremos en la sección 4, el uso de Kerberos será la solución al ataque de MITM y presentaremos que si existen formas

de enviar tickets de Kerberos en redes LAN, WAN y hasta redes que operen tanto con IPv4 e IPv6.

En la siguiente sección presentaremos la explicación de los tipos de ataques en IPv4 e IPv6, que fueron mencionados al inicio del trabajo.

2.6. Ataques a las redes en IPv4 e IPv6

Hemos mencionado en el planteamiento del problema de este proyecto, que existen algunos ataques a IPv4 e IPv6, ataques que con los años se están mitigando, y con el tiempo se tendrán que corregir de forma permanente.

En este punto nos centraremos en lo que es el ataque de "hombre en el medio" daremos una breve explicación sobre los distintos tipos de "hombre en el medio" en redes IPv4 e IPv6:

A. MITM sobre redes IPv4 utilizando ARP spoofing.

Primero y de acuerdo con [Pat13] el conocido "Man In The Middle" es un tipo ataque, que consiste en un tercero el cual creara la posibilidad de leer, inyectar o modificar la información presente en un canal entre 2 host o equipos, sin que ninguno de ellos se entere. Ahora entre estos ataques de MITM en IPv4 e IPv6, el ataque de MITM en redes IPv4 basado en spoofing con ARP, consiste en:

[Pat13] Enviar mensajes ARP a la red Ethernet, esto se realiza normalmente con la finalidad de asociar la dirección MAC del atacante con la dirección IP de otro equipo. Cualquier tráfico dirigido a la dirección IP de la puerta de enlace predeterminada será erróneamente enviado al atacante, en lugar de a su destino real.

B. MITM sobre redes IPv4 utilizando inyecciones DHCP ACK.

Este tipo de ataque, como nos lo presenta [Pat13] consiste en un atacante que monitoriza los intercambios DHCP y en un determinado punto de la comunicación envía un paquete para modificar su comportamiento.

C. MITM sobre redes IPv6 utilizando neighbor advertisement spoofing.

El principio de este ataque, como lo presenta [Pat13] es el mismo que el de ARP spoofing, con la diferencia en que IPv6 no trabaja con el protocolo ARP, sino que toda la información se transmite a través de paquetes ICMPv6. Actualmente existen 5 tipos de paquetes ICMPv6 utilizados en el protocolo de descubrimiento.

D. MITM sobre redes IPv6 utilizando SLAAC attack.

El objetivo de este ataque, de acuerdo con [Pat13] es poder hacer un MITM cuando un usuario se conecta a internet a un servidor que no tiene soporte para IPv6 y que por lo tanto es necesario conectarse usando IPv4. Este ataque puede ser resuelto al encargarse de la resolución de nombres de dominio una vez situada en el medio de la comunicación, transformando las direcciones IPv4 en direcciones IPv6.

E. MITM sobre redes IPv6 utilizando falso DHCPv6.

Este ataque, tal y como lo indica [Pat13] consiste en que el atacante se hace pasar por el servidor DCHPv6, respondiendo a todas las solicitudes de la red, repartiendo direcciones IPv6 y un DNS falsos para manipular el destino de los usuarios o denegar el servicio.

2.7. Internet security association and key management protocol (ISAKMP)

En esta sección presentaremos un punto importante para la autenticación, se trata de la asociación de seguridad de internet y el protocolo de administración de claves, mejor conocido por sus siglas ISAKMP. Este concepto además de una asociación es un protocolo muy importante, el cual combina los conceptos de seguridad de autenticación, gestión de claves y asociaciones de seguridad (SA), para establecer la seguridad necesaria en las comunicaciones gubernamentales, comerciales y privadas en internet.

Como se presenta en el estándar [MS98], el ISAKMP generalmente utiliza el protocolo IKE para su intercambio de claves, cabe mencionar que también puede utilizar otros métodos. ISAKMP define los procedimientos y formatos de paquetes para establecer, negociar, modificar y borrar asociaciones de seguridad (SA). Estas SAs contienen toda la información necesaria para la ejecución de diversos servicios de seguridad de red, servicios tales como los servicios de la capa IP, servicios de la capa de transporte o aplicación, entre otros servicios.

Ya conocemos que toda información que se envía por la red tiene una carga útil o payload, y como se muestra en [MS98] el ISAKMP define esas cargas útiles para el intercambio de la generación de claves y la autenticación de datos. Estos formatos proporcionan un marco coherente para la transferencia claves y datos de autenticación, los cuales son independientes de la técnica de generación de clave, algoritmos de cifrado y mecanismos de autenticación.

Según [MS98], el ISAKMP separara los detalles de la gestión de las SAs (y la gestión de claves), de los detalles del intercambio de claves. En el mundo de las telecomunicaciones, puede haber muchos protocolos de intercambio de claves. Sin embargo, se requiere de un marco común que sirva tanto para acordar el formato los atributos de las SAs, para negociar, modificar y eliminar las SAs. Es precisamente el ISAKMP quien sirve como este marco común.

2.7.1. Las cargas útiles del ISAKMP

Tal y como lo señala [MS98], las cargas útiles del ISAKMP, nos proporcionan los bloques de construcción modulares que se utilizarán para la construcción de los mensajes ISAKMP. La presencia de las cargas útiles ISAKMP, así como su ordenamiento se definen y dependerán de un campo llamado "tipo de intercambio", este campo es uno de los tantos campo de la cabecera de ISAKMP, la cual mostramos a continuación.

2.7.2. La cabecera del ISAKMP

De acuerdo con [MS98], todo mensaje ISAKMP tendrá un formato de cabecera fijo, seguida de un número variable que identificará a las cargas útiles. Se utiliza el tipo de cabecera fija mostrada en la Figura 2.9, debido a que una cabecera fija simplifica el análisis, beneficiando el análisis de protocolos para softwares, que serán menos complejos y más fáciles de implementar.

<table>
<tr><th></th><th>1 Byte</th><th colspan="2">2 Byte</th><th>3 Byte</th><th>4 Byte</th></tr>
<tr><th></th><th>0 – 7 bits</th><th colspan="2">8 – 15 bits</th><th>16 – 23 bits</th><th>24 – 31 bits</th></tr>
<tr><td>1
1</td><td colspan="5">Iniciador Cookie</td></tr>
<tr><td>1
1</td><td colspan="5">Respondedor Cookie</td></tr>
<tr><td>1
1</td><td>Siguiente Carga Útil</td><td>Versión Principal</td><td>Versión Secundaria</td><td>Tipo de Intercambio</td><td>Banderas</td></tr>
<tr><td>1</td><td colspan="5">ID del Mensaje</td></tr>
<tr><td>1</td><td colspan="5">Tamaño</td></tr>
</table>

Figura 2.9: Cabecera del ISAKMP. [MS98]

La cabecera fija contiene la información requerida por el protocolo para mantener el estado, las cargas útiles de proceso y posiblemente la información para prevenir la denegación de servicio o ataques de repetición.

Hemos hablado del campo de la cabecera ISAKMP, llamado “tipo de intercambio”, como afirma [MS98] este campo, son importantes y necesarios de conocer, es por ello que los explicaremos a continuación:

- El iniciador de cookie (8 octetos): Este campo inicia el establecimiento de las SAs, las notificaciones o las eliminaciones de las SAs.
- El responder de cookie (8 octetos): Este campo responde a las solicitudes de establecimiento de las SAs, de las notificaciones o las eliminaciones de las SAs.
- La siguiente carga útil (1 octeto): Este campo indica el tipo de la primera carga útil del mensaje.
- La versión principal (4 bits): Este campo, muestra la versión principal del protocolo ISAKMP que está en uso en dicho momento.
- La versión secundaria (4 bits): Este campo muestra la versión menor del protocolo ISAKMP que se estará utilizando.
- El Ttpo de intercambio (1 octeto): Este campo indica el tipo de intercambio utilizado, estos indican los ordenamientos de los mensajes y de la carga útil en los intercambios ISAKMP.
- Las banderas (1 octeto): Indican las opciones específicas que se utilizarán y establecerán para el intercambio ISAKMP.
- El ID del mensaje (4 octetos): Es un mensaje utilizado con el fin de identificar el estado de protocolo durante la segunda fase de las negociaciones.
- El tamaño (4 octetos): Este campo comprende la longitud total del mensaje.

3 Estado del arte

En este punto se presentan los últimos trabajos e investigaciones relacionadas con el tema que estamos tratando, estos trabajos presentan y analizan muchos de los aspectos que estamos estudiando, de manera que sirven como base para el desarrollo y diseño de la nueva forma mejorada de autenticación para IPV6. Debido a que IPv6 es un tema muy amplio, dividiremos los temas de la siguiente manera:

3.1. El protocolo de internet, el agotamiento de direcciones y la transición de IPv4 a IPv6

3.1.1. El protocolo de internet

Un protocolo de Internet (IP) es un protocolo de la capa de red, el cual define como se comunican los dispositivos a través de las redes, es decir que este protocolo se utiliza en sistemas interconectados de redes de comunicación computacional de conmutación de paquetes. Actualmente hay dos tipos de protocolos de internet en uso activo, IP versión 4 (IPv4) e IP versión 6 (IPv6) y cuyas direcciones IPv4 e IPv6 generalmente se asignan de forma jerárquica. Después de examinar los trabajos realizados por [Yan14, Cal, yRSA16], podemos concluir que independientemente de sus ventajas, direccionamientos, arquitecturas y mejoras que se puedan realizar en cada uno, existe un concepto el cual se debe considerar como el más crítico sobre la IP, el cual es el siguiente:

3.1.2. El agotamiento de direcciones

El término agotamiento de direcciones IPv4 se refiere a la etapa de reservas donde las asignaciones se restringen en tamaño y periodicidad. Este agotamiento implica que no se tendrá suficientes direcciones para cubrir las necesidades de direccionamiento IPv4. Habiéndose estudiado los trabajos de [yRSA16, Cas16, Mex16] podemos resumir 2 alternativas:

- Optar por el uso del IPv6.
- Hacer uso de restricciones y políticas para la solicitud de espacio adicional de direcciones IPv4.

Sin embargo esta última supondría un mercado y un costo elevado para la adquisición de direcciones IPv4

3.1.3. La transición de IPv4 a IPv6

Tomando los alcances de [yRSA16, yFR13] es necesario concluir que los protocolos Ipv4 e Ipv6 no pueden trabajar juntos, debido a que no es posible efectuar una coexistencia entre ambos, por lo que la solución es la transición de Ipv4 a Ipv6. Finalmente recopilando las enseñanzas del profesor [Cal] y [Yan14] podemos concluir las siguientes ventajas de IPv6 sobre IPv4: :

- La falta de direcciones de IPv4, hace que ya no podamos depender de IPv4. IPv4 tiene un espacio de direcciones de 32 bits, en cambio IPv6 ofrece un espacio de 128 bits, con ello tiene más niveles de jerarquías de direccionamiento y más nodos direccionables.
- IPv4 posee enormes tablas de ruteo en el backbone de Internet, lo que lo hace ineficaz y perjudica los tiempos de respuesta.
- Debido a la multitud de nuevas aplicaciones en las que IPv4 es utilizado, ha sido necesario agregar nuevas funcionalidades al protocolo básico, aspectos que no fueron contemplados en el análisis inicial de IPv4, lo que genera complicaciones en su escalabilidad para nuevos requerimientos y en el uso simultáneo de dos o más de dichas funcionalidades. Entre las más conocidas se pueden mencionar medidas para permitir la calidad de servicio (QoS), seguridad y movilidad.
- El IPv6 permite la conexión de millones de dispositivos con capacidad IP, que siempre están en funcionamiento y cada uno de ellos teniendo su propia y exclusiva dirección IP.
- Otra de las ventajas que parte del mayor espaciamiento de direcciones, es que IPv6 tiene un enorme espacio y capacidad para direcciones IP, con ello tiene mayor seguridad incorporada y características de movilidad, “plug-andplay” (conecte y haga funcionar) hasta auto-configuración de direcciones, renumeración simplificada del sitio y redes y servicios de fácil re-diseño.

3.2. IPv6 y sus puntos a solucionar

En el apartado anterior hemos presentado los alcances de la IP, y se concluyó con la notable opción de optar por IPv6 por encima del IPv4. Los trabajos presentados en este apartado [Ros16, TEC14, Cal]nos muestran los alcances del IPv6, los cuales son importantes ya que nos introduce con los conceptos de limitaciones o problemas en su seguridad que presenta IPv6, de forma específica encontramos que existen distintos tipos de ataques a IPv6, que podrían agruparse de la siguiente forma:

- Intromisión: Acceso no autorizado, cracking de claves WiFi.
- Espionaje: Wardriving, ingeniería social.
- Interception: Man in the middle.
- Suplantación: Phishing, ARP spoofing, IP spoofing.
- Modificación: SQL Injection, XSS, CSRF.
- Denegación de servicio: DoS/DDoS.

Sin embargo y como nos lo revelán los trabajos de [Ros16, TEC14] estamos hablando de un protocolo que puede mejorarse lo que nos lleva a seguir optando por esta tecnología y trabajar en mejorar sus puntos débiles. Primero habiendo estudiado todos los artículos referentes a IPv6, se observó que IPv6 presenta un uso obligatorio del IPsec, a diferencia de IPv4 que hace uso de este protocolo de forma opcional, también se pudo observar que en los distintos tipos de ataque a IPv6, muchos de estos fueron para IPv4, lo que demuestra incluso una migración en cuanto a ataques. Y en los últimos años, la mayoría de los ataques a las redes, han sido del tipo "hombre en el medio".

Precisamente estos ataques de hombre en el medio se basan en la interceptación de tráfico entre dos entidades, debido a que cada uno de los equipos de una red tiene una dirección IP y existen las tablas ARP, se puede concluir que se tiene una memoria caché la cual puede resolver direcciones IP a MAC de forma más rápida y ayudar a que los paquetes lleguen a su destino. En resumen con una interceptación MitM se puede fácilmente atacar a las tablas ARP, asignándoles la dirección MAC de un intruso a la dirección IP de un objetivo, de esta forma un router interpretará que el tráfico va dirigido correctamente.

Englobando todos los trabajos de [Ros16] y analizando todos los puntos a solucionar en IPv6, se tienen 3 aspectos que se deben mejorar en IPv6, dado que este protocolo mejorable tiene problemas en:

- La autenticación en IPv6.
- La encriptación en IPv6.
- La integridad en IPv6.

Cada uno de estos aspectos debe mejorarse y el que presenta un menor enfoque es la autenticación, muchos trabajos sobre mejoras a IPv6 no se enfocan en la autenticación y para desarrollar este punto se parte del IPsec.

3.3. IPsec es la solución

Se han desarrollado investigaciones y pruebas en IPv6, con el fin de determinar la manera de mejorar este protocolo, trabajos sobre las vulnerabilidades y ataques en IPv6 mostrados en [Fra09] y los artículos del profesor García en [Gar12a], nos llevan a concluir que IPsec es la clave para garantizar mayor seguridad en IPv6.

Y esta premisa se debe a que tal IPsec proporciona servicios de seguridad en la capa de red, específicamente y como nos presenta el trabajo [Fra09], IPsec le permite a un sistema seleccionar los protocolos de seguridad, así como determinar los algoritmos a utilizar, e implementar cualquier algoritmo criptográfico que requiera con el fin de proporcionar los servicios solicitados.

Las técnicas de "Fuzzing" y "Packet Crafting", vistas en el trabajo [Gar12a] realizan la modificación de paquetes de red para la comprobación del funcionamiento de los distintos elementos de red, y las pruebas en escenarios específicos, nos llevan a concluir que si bien IPv6 presenta notables retos y los problemas en los mecanismos de control de acceso basados en direcciones IP y riesgos de privacidad derivados de la desaparición de NATs, son existentes, la solución es el uso de los mecanismos de seguridad que ofrece IPv6, es decir IPsec. Las pruebas en Scapyst, muestran vulnerabilidades que pueden ser explotadas por usuarios maliciosos, y precisamente mediante IPsec se puede mejorar y adaptar al protocolo IPv6.

3.3.1. IPsec y las claves criptográficas

Es importante mencionar los aportes de los trabajos de [Fra09, Gar12a], sobre la relación entre IPsec y los distintos sistemas que pueden trabajar en conjunto, y es que los servicios de seguridad que ofrece IPsec, usan valores secretos compartidos (es decir claves criptográficas), y es que IPsec utiliza un conjunto de mecanismos adicionales, para colocar estas claves en su sitio. El documento [Fra09] nos introduce y detalla la existencia de la distribución manual y automática de claves, de hecho existe un método basado en clave pública (IKE – ISAKMP) para la gestión automática de claves, e incluso IPsec permite técnicas de distribución automatizadas con sistemas, tales como los basados en KDC de Kerberos, protocolo que utilizaremos en nuestra propuesta.

En resumen existe una compatibilidad entre IPsec y otros protocolos, y precisamente se aprovechará esta compatibilidad para el desarrollo de IPv6 complementado con Kerberos, antes de culminar este punto es necesario tomar en cuenta la siguiente premisa, planteada por los trabajos de [Fra09, Ros16], IPsec y los algoritmos que estén asociados al mismo, permiten proporcionar seguridad de alta calidad para el flujo de

tráfico de Internet, como ya lo hemos mencionado, sin embargo la seguridad ofrecida por estos protocolos depende en última instancia de la calidad de implementación de estos protocolos.

3.4. ISAKMP, IKE y el uso de claves

En el apartado anterior se mencionó que IPsec permite la distribución de claves, con métodos basados en IKE – ISAKMP y también con sistemas que utilicen un tercero como el KDC de Kerberos. Y precisamente si existe un método para IPsec utilizando IKE, porque no utilizarlo. La respuesta radica en la utilización de claves.

Primero los aportes de los trabajos de [Fra09, TEC14, Gar12a], no revelan que ISAKMP combina los conceptos de seguridad de autentificación, administración de claves y de asociaciones de seguridad (SA) para establecer el tipo seguridad que se necesite, y son estas SAs, las cuales contienen toda la información requerida para la ejecución de diversos servicios de seguridad de red.

Es así que ISAKMP proporciona un marco para la autentificación y el intercambio de claves, en cuanto al IKE, el documento de [Fra09, KS12] nos enseña que, este es un protocolo el cual al ser usado conjuntamente con ISAKMP permite obtener material autenticado para usarse con ISAKMP, y para otras SAs que se negocian en ciertos servicios.

En conclusión con el protocolo IKE es posible negociar las SAs de un modo protegido, y como se observa en los trabajos de [Fra09, TEC14] toda esta tarea es posible gracias a los siguientes atributos que pertenecen:

- Algoritmo de encriptación,
- Algoritmo de hash.
- Método de autentificación.
- Información sobre un grupo al cual realizarle Diffie Hellman.

¿Y por qué utilizar claves?

La experiencia y los estudios nos muestra que para garantizar la autenticación se puede usar claves (passwords), incluso cuando se envía una clave para realizar la autenticación, esta clave nos ayuda a identificar la identidad del usuario, sin embargo muchos estudios revelan que si bien enviar claves ayuda en la autenticación, esto

resulta inseguro debido a que se abre la posibilidad que un tercero puede usar estas claves para robar la información.

¿Entonces cómo solucionar este problema?

La respuesta planteada por [TEC14, Gar12a, KS12] es el uso de un hash sobre una clave, de esta forma un emisor creará una cadena de caracteres y el receptor comprobará la autenticidad de la clave encriptada mediante su propio cálculo de hash en su lado del receptor, sin embargo un hash no es un proceso regresivo, por lo que esta técnica solo permitirá autenticar las claves, pero no será posible descifrar estas claves para poder utilizarlas. Por lo que después de analizar todos estos trabajos y aportes se llegó a la conclusión que la respuesta es el no uso de claves, es aquí donde entra Kerberos y el segundo método que se mencionó al inicio de este apartado.

3.5. Kerberos e IKE

Hasta este punto hemos demostrado el inconveniente de utilizar claves, si bien se puede lograr una autenticación con el método de integración de IPsec con IKE – ISAKMP, sin embargo a pesar que la metodología de contraseñas es muy conveniente, el tráfico de contraseñas es riesgoso.

El uso del protocolo Kerberos es la respuesta, debido a que opera con un KDC para la gestión de tickets, los estudios mostrados en [AD00] presentan que el protocolo Kerberos garantiza un excelente nivel de autenticación, además de ser compatible y adaptable con IKE, como se ve en [LB08].

Si bien Kerberos no utiliza passwords, debido a que su planteamiento muestra que es innecesario, eso no significa que Kerberos no pueda hacer uso de estos.

Los trabajos de [AD00, Fra09] nos presentan los alcances de Kerberos, el cual utiliza la criptografía de claves simétricas, igual que hash, para validar a los usuarios. Kerberos trabaja sobre la base de "tickets", los cuales son los encargados de demostrar la identidad de los usuarios, estos tickets actúan como identificadores de identidad.

En resumen, la nueva forma de autenticación busca que la autenticación en IKE utilice un password de autenticación, el cual se encontrará protegido por un Hash y dicho password se le adicionará a Kerberos, dado que una forma de autenticar en IKE es con el Kerberos, de esta forma se crea una mejora de autenticación en IPsec y en IPv6.

3.6. Conclusiones

De acuerdo a los trabajos previamente mostrados, podemos concluir lo siguiente:

- Primero el protocolo IPv6, pese a sus ventajas y desventajas, es un protocolo mejorable y adaptable, por lo que se puede hacer de este protocolo, la base para ofrecer comunicaciones mucho más seguras.
- Segundo el protocolo IPv6, pese a sus estudios y avances frente a la seguridad, no es muy estudiado en las formas de como realiza su autenticación.
- Tercero la base para el desarrollo de nuevas formas de autenticación y de seguridad en IPv6, parte del estudio y mejoramiento del IPsec, porque como se ha concluido en el primer trabajo, IPsec es el único camino para hacer de IPv6 el protocolo más seguro.
- Cuarto ya no se puede seguir dependiendo del IPv4, porque ya está demostrado que IPv4 tiene un tiempo de cierre, por lo que IPv6 es el protocolo que se debe de seguir desarrollando.
- Finalmente no existe un planteamiento de una forma mejorada para la autenticación del IPv6, y de forma específica no existe un estudio de bit a bit en la forma en que IPv6 realiza la autenticación.

4 Propuesta de solución

En esta sección se detallará a profundidad el funcionamiento de la nueva propuesta para la autenticación en IPv6, propuesta que solucionará el ataque de MITM que se da en las redes IPv6. Para la siguiente explicación a profundidad, se plantean ciertos escenarios, donde se detallará el modo de funcionamiento de la nueva propuesta, para dichos escenarios.

4.1. Comunicación en una red LAN

Para este primer escenario, se tiene una comunicación en una red LAN entre dos usuarios, a los cuales nombraremos "A" y "B". Esta comunicación puede darse también entre un usuario con un servidor o un servidor con otro servidor, pero la comunicación se dará en una topología LAN.

En la Figura 4.1 se presenta la topología LAN, en la que se puede apreciar a los dos usuarios que se comunicarán, el KDC de Kerberos que se encontrará en la Internet (la cual se la presenta como una nube), el router/switch que se encuentra en el medio y las conexiones. No detallaremos que tipo de conexiones o cables se utilizaran, se asume que se conoce como realizar una red LAN básica.

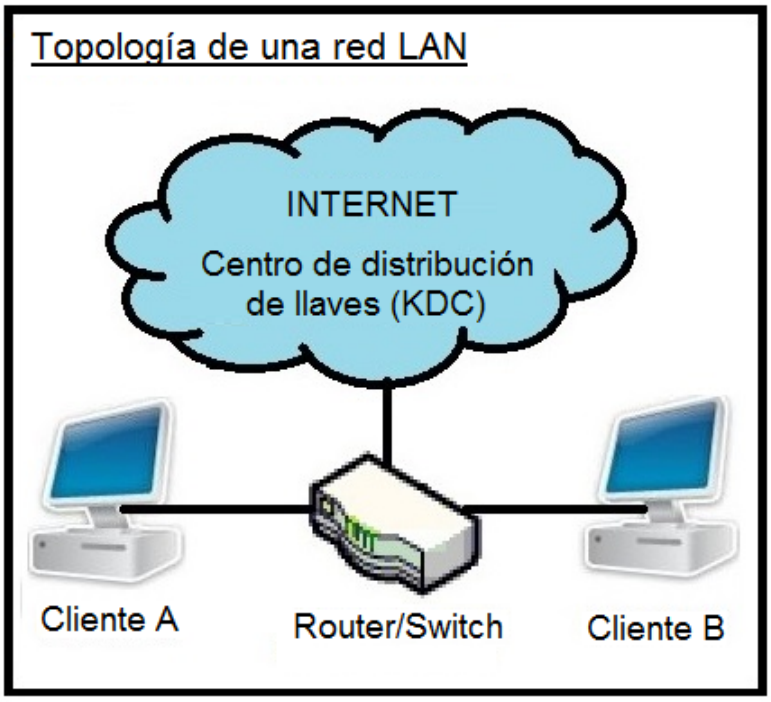

Figura 4.1: Topología en una red LAN.

Como se observa en la Figura 4.1, la topología LAN muestra una comunicación entre los dos clientes con la Internet y entre clientes. La propuesta planteada para una red LAN, consiste en una comunicación entre clientes y el KDC que se encuentra en la Internet, de la misma forma que el protocolo Kerberos, pero se realizará en 3 pasos:

- Él Registro de los Clientes con el KDC, con Kerberos.
- Él Envío de Tickets a los Clientes, con Kerberos.
- Él Envío de la Información entre clientes, con IPv6.

A. Registro de los clientes al KDC.

Entonces en el primer punto se realizara el registro de ambos clientes al servidor KDC, con ayuda de Kerberos, de esta forma se hará uso de la autenticación del tipo Kerberos, para garantizar la identidad de los clientes.

Como se puede observar en la Figura 4.2, se comienza con la autenticación de tanto los Clientes “A” y “B” al servidor KDC, este proceso de autenticación y registro se realiza mediante un proceso de logueo de los clientes. Durante este proceso de

registro de clientes, tanto “A” como “B” enviaran un paquete con un mensaje de identificación al KDC con sus nombres y la petición deseada, en caso de haber una, este mensaje estará protegido por sus respectivas llaves privadas de usuarios.

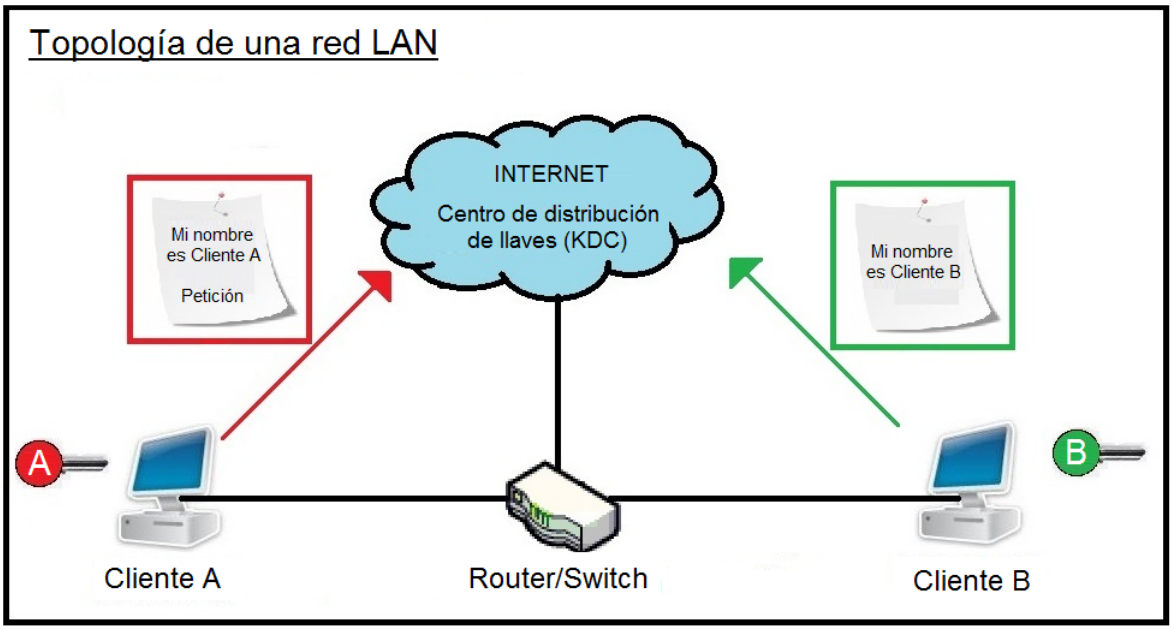

Figura 4.2: Registro de los clientes al KDC.

De forma que un intruso no podrá leer dichos mensajes con facilidad, y de hacerlo solo encontrará el nombre del usuario, más no encontrará ningún password o clave secreta, dado que no se ha enviado dicha información. Estos mensajes de registro solo contendrán los nombres de los Clientes y la petición del Cliente, en caso de necesitar de una petición, al servidor KDC.

B. Primera corroboración de la identidad de los clientes.

Como se observa en la Figura 4.3, siendo los clientes “A” y “B” clientes Kerberos, el servidor KDC tendrá una base de datos con dichos clientes y podrá abrir dichos mensajes, porque tendrá acceso a las llaves privadas en su base de datos, es decir que en la base de datos del KDC, este servidor conoce tanto a los clientes “A” y “B”, de manera que cuando el KDC reciba los paquetes de registro de los clientes, el servidor KDC podrá corroborar la identidad de los clientes con su base de datos, corroborando tanto las IPs de clientes de los clientes, con las IPs de las bases de datos que contiene, y los nombres de los usuarios que recibió en los paquetes, con los nombres que tiene en base de datos.

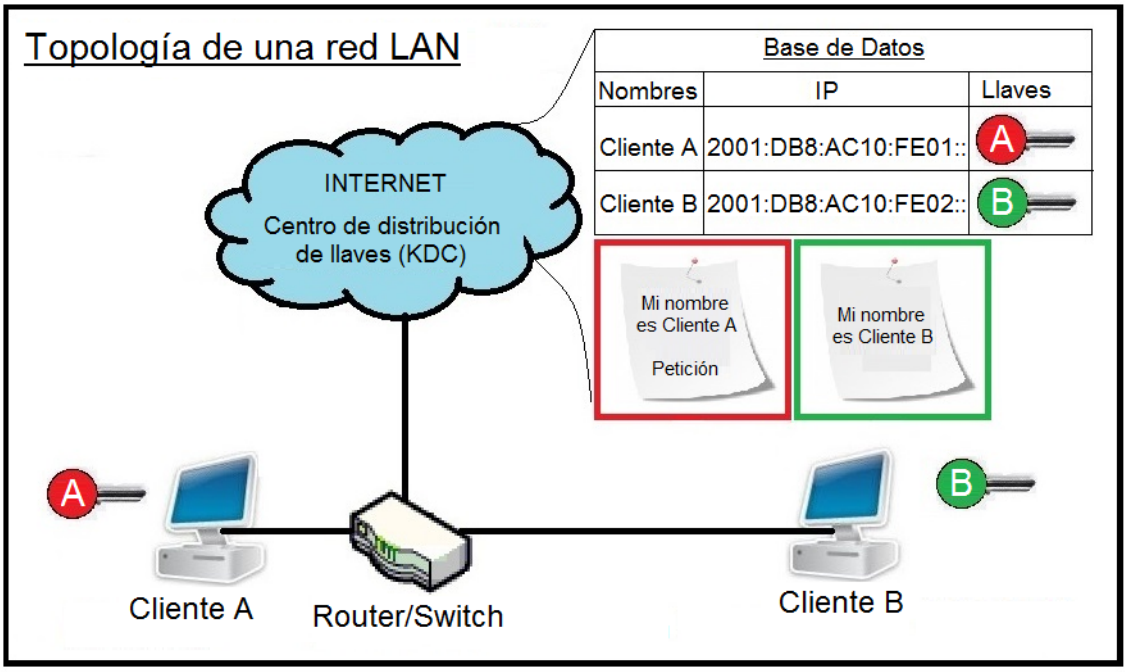

Figura 4.3: Primera corroboración de la identidad de los clientes.

C. Envío de los TGTs a los clientes

De haber corroborado la identidad de los usuarios, el servidor KDC generará y enviará un TGT a cada uno de los usuarios, dicho TGT tendrá la misma llave de sesión, para asegurar la comunicación entre los dos clientes o para asegurar la petición deseada. Dichos TGTs tendrán una línea de tiempo específico, como parte de su seguridad y una vez agotado dicho tiempo el ticket será destruido.

Entonces el servidor KDC genera dos paquetes, en los cuales ingresa el TGT, protegido por una llave privada, la cual solo la conocerá el KDC, y por lo tanto ninguno de los clientes o un intruso podrá leer el contenido del TGT, el cual contendrá la siguiente información:

- El nombre del cliente.
- El nombre del KDC.
- La IP del cliente.
- La llave de sesión.

Como se observa en la Figura 4.4, además del TGT, en dicho paquetes el KDC incluirá una copia de la llave de sesión del TGT generado, y dichos paquetes se protegerán con las respectivas llaves privadas de los Clientes y se enviaran. De ser

los mensajes interceptados por intrusos, la información no podrá ser alterada ni leída, gracias a la doble protección de llaves en cada paquete. Cabe mencionar que estas llaves de sesión (Session Key) son llaves de encriptación con líneas de tiempo, creadas exclusivamente para la presente sesión, la cual cambiara progresivamente, como medida de seguridad.

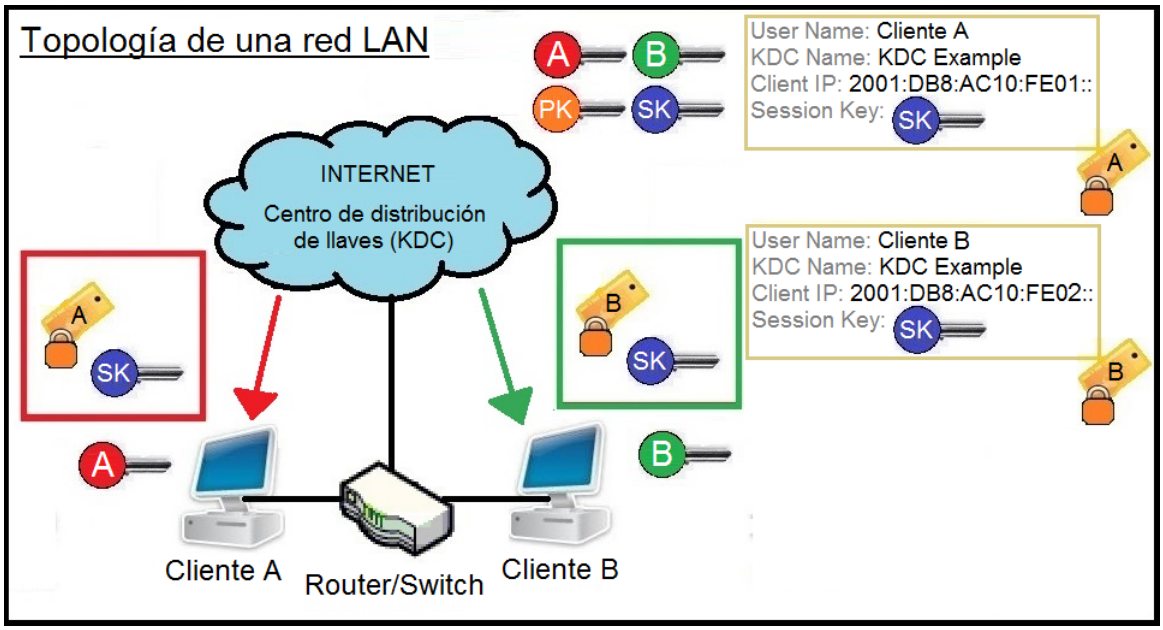

Figura 4.4: Envío de los TGTs a los clientes.

D. Envió de los TGTs y los autenticadores.

Continuando y como se observa en la Figura 4.5, los cliente Kerberos, des-encriptarán los paquetes recibidos con sus llaves privadas y recibirán tanto los TGTs, como la copia de llave de sesión. En estos momentos los clientes poseen una llave de Sesión, la cual utilizarán para los próximos procesos. Ambos clientes crean un "Autenticador", el cual contendrá:

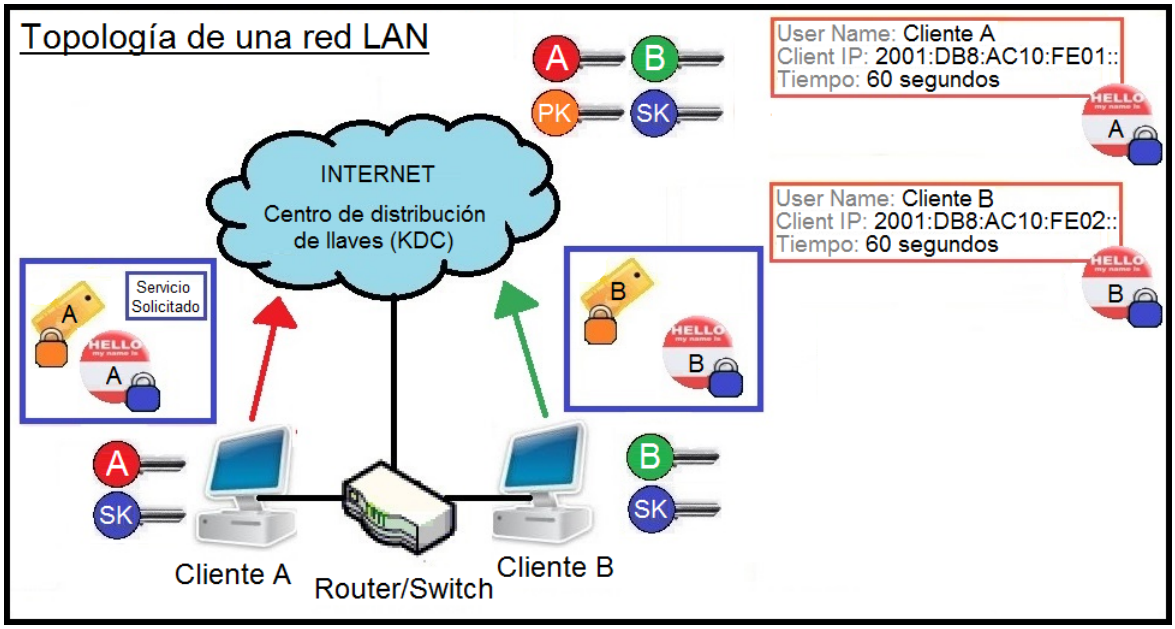

Figura 4.5: Envió de los TGTs y los autenticadores.

- El nombre del cliente.
- La IP del cliente.
- El tiempo.

Estos autenticadores serán encriptados por los clientes con las llaves de sesión que recibieron y enviaran cada uno un paquete nuevamente al KDC, con "el autenticador encriptado", más el "TGT" que recibieron y el nombre del "servicio o petición que se deseen solicitar". Dicho paquete se encriptara de forma adicional con nuevamente la llave de Sesión que recibieron ambos clientes, de forma que también se tiene una doble protección de llaves en estos paquetes.

E. Recepción del ticket de servicio y la nueva llave privada del KDC.

A continuación, el servidor KDC recibirá los mensajes y podrá des-encriptarlos con la llave de sesión que posee, además podrá verificar el TGT utilizando su llave privada, para corroborar que el TGT no haya sido abierto ni alterado, de comprobar la integridad del TGT, el KDC verificara nuevamente la identidad de los clientes con su base de datos y los autenticadores que recibió, corroborando las IPs y los nombres respectivos.

De ser positiva la identidad de los clientes, el KDC creara un nuevo ticket, pero solo para el cliente que solicite un servicio o tenga una petición, el otro cliente no

recibirá el ticket porque se asumirá que dicho cliente será quien resuelva la petición. El nuevo ticket se conoce como "ticket de servicio", este ticket será exclusivo para el servicio o petición deseada. El ticket de servicio también presentara una línea de tiempo y contendrá la siguiente información:

- El nombre del cliente.
- El servicio o petición solicitada.
- La IP del cliente.

Una nueva llave de sesión. Como se observa en la Figura 4.6, el servidor KDC encriptara el ticket de servicio con una "llave privada de servicio", de forma similar al caso anterior, por lo que los clientes no podrán acceder a la información de los tickets de servicio. El servidor KDC creara un nuevo mensaje cuyo contenido será el "ticket de servicio encriptado con la nueva llave privada de servicio", y una copia de la "nueva llave de sesión, y dicho paquete se encriptara con la antigua llave de sesión, para ser enviado al cliente que mando la solicitud o petición.

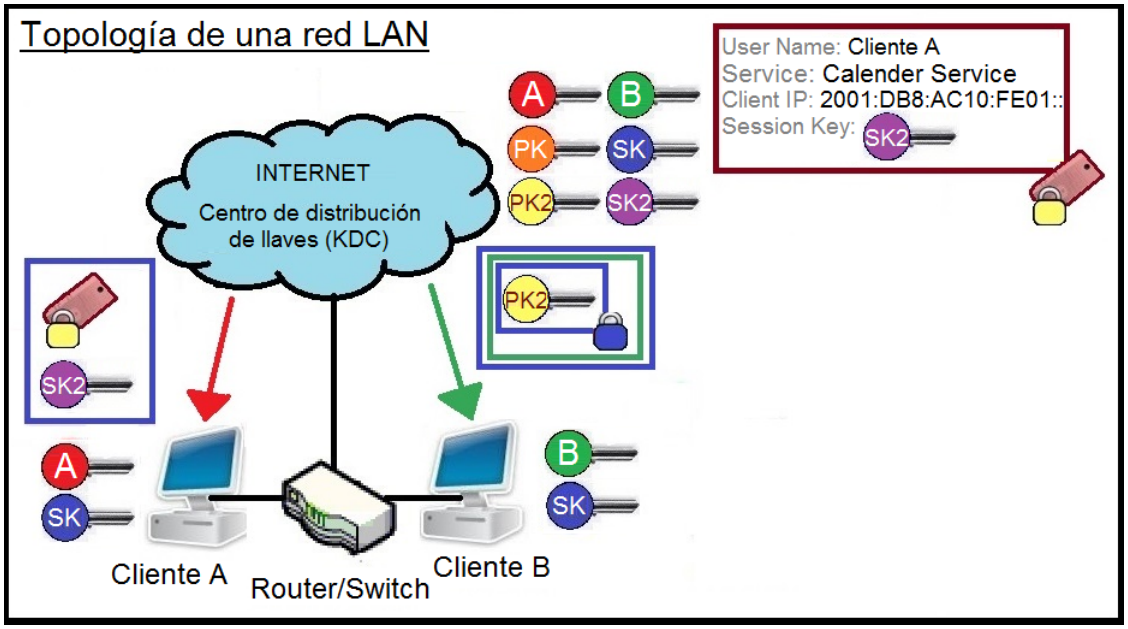

Figura 4.6: Recepción del ticket de servicio y la nueva llave privada del KDC.

Mientras que el cliente que no envió la solicitud, se le enviará un paquete con un mensaje conteniendo la nueva llave privada del servidor KDC, la cual se usará para abrir el ticket de servicio, dicho mensaje con la llave se protegerá con la antigua llave de sesión y todo el paquete también se protegerá con llave privada del usuario y sobre esta se la volverá a proteger con la antigua llave de sesión.

F. Inicio de la comunicación entre clientes.

A continuación y como se muestra en la Figura 4.7, los clientes nuevamente podrán des-encriptar los mensajes recibido con la antigua llave de sesión, recibiendo uno de ellos el ticket de servicio y la copias de la nueva llave de sesión, mientras el otro cliente recibirá la copia de la llave privada para abrir el ticket de servicio, a partir de este momento los clientes dejan de comunicarse con el KDC y procede a comunicarse entre ellos.

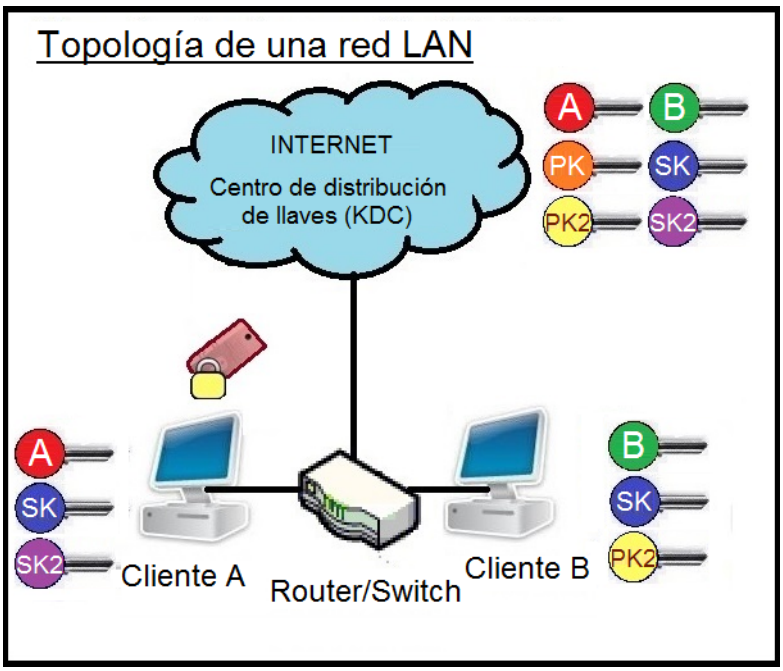

Figura 4.7: Inicio de la comunicación entre clientes.

G. Envió del paquete IPv6 del primer cliente, al segundo cliente.

Como se observa en la Figura 4.8, el primer cliente creara un nuevo autenticador y lo encriptara con la nueva llave de sesión recibida del KDC, luego procede a enviar un paquete al segundo cliente, con tanto el "autenticador encriptado" y el "ticket de servicio protegido" que recibió del KDC, este paquete será enviado con IPv6.

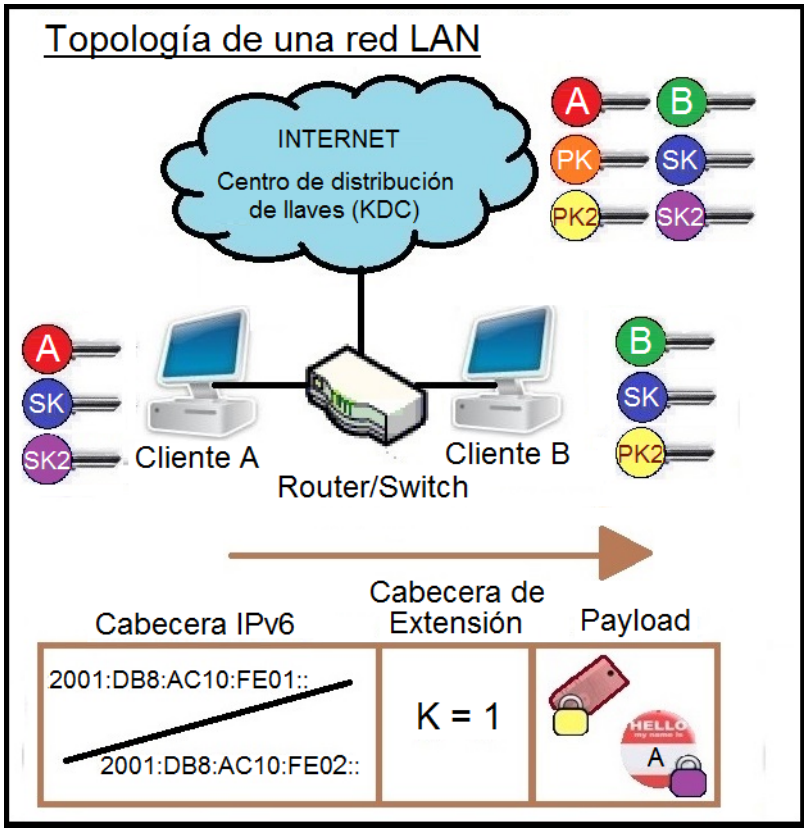

Figura 4.8: Envió del paquete IPv6 del primer cliente, al segundo cliente.

Como todo paquete IPv6, en su cabecera se ingresaran las IPs en IPv6 del emisor y del receptor, en la cabecera de extensión se enviara un Flag (bandera), la cual servirá para indicar que en el payload del IPv6 o datos del IPv6 se está enviando datos de Kerberos, que al final son datos, y finalmente en el payload o carga de datos se enviara el paquete creado por el usuario, con sus llaves protegiendo los datos.

H. Envió del paquete IPv6 del segundo cliente, al primer cliente.

Finalmente y como se observa en la Figura 4.9, el segundo Cliente recibirá el paquete IPv6 y utilizando la copia de la llave privada de servicio que recibió del KDC, podrá des-encriptar el ticket de servicio. Sabiendo el cliente Kerberos que la información del ticket fue generada por el KDC, el segundo cliente utilizara la nueva llave de sesión que se encuentra dentro del ticket de sesión, para acceder al autenticador que recibió.

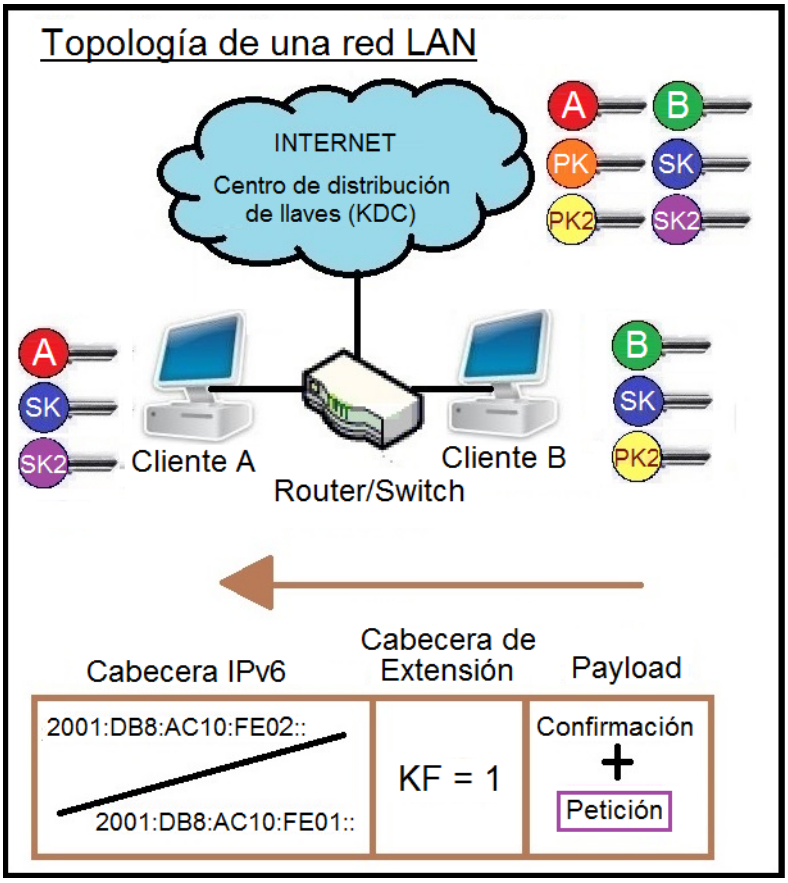

Figura 4.9: Envió del paquete IPv6 del segundo cliente, al primer cliente.

Así el segundo cliente podrá confirma la identidad del primer cliente utilizando el nombre y la IP que aparecen en el ticket de servicio y en el autenticador, y de ser positiva la identidad del primer cliente, el segundo cliente enviara un paquete IPv6 con las IPs del emisor y receptor en la cabecera, un Flag en la cabecera de extensión indicando que la información del Payload no es Kerberos y en el Payload enviara una paquete con una confirmación, más el servicio o petición resuelta, esta estará protegida por la nueva llave de sesión.

4.2. Comunicación en una red WAN

Para este segundo escenario, presentaremos una comunicación en una red WAN entre tres usuarios, a los cuales nombraremos “A”, “B” y “C”, la razón por la que incluimos un usuario extra, a diferencia de la topología anterior, es para demostrar que el proceso se repetirá aun para un usuario más, es decir que la propuesta no se limita a un máximo de dos usuarios, sino que puede utilizarse para más de dos

usuarios.

Incluso la topología de la red LAN puede usarse para más de dos usuarios, dado que el proceso es el mismo para todos los usuarios que se desee incluir. Volviendo a la topología de la red WAN, esta comunicación puede darse también entre usuarios con un servidores o servidores con otro servidores.

En la Figura 4.10, se presenta la topología WAN, en la que se puede apreciar a los tres usuarios, ubicados cada uno en una red diferente, dado que es una red WAN, existe una conexión entre estas tres diferentes redes, pero que al final son redes de internet, y por ello cada red Internet, tendrá un distinto KDC, además de los routers que se encuentra en el medio. Nuevamente no detallaremos que tipo de conexiones o cables se utilizaran, dado que asume que se conoce como realizar una red WAN básica.

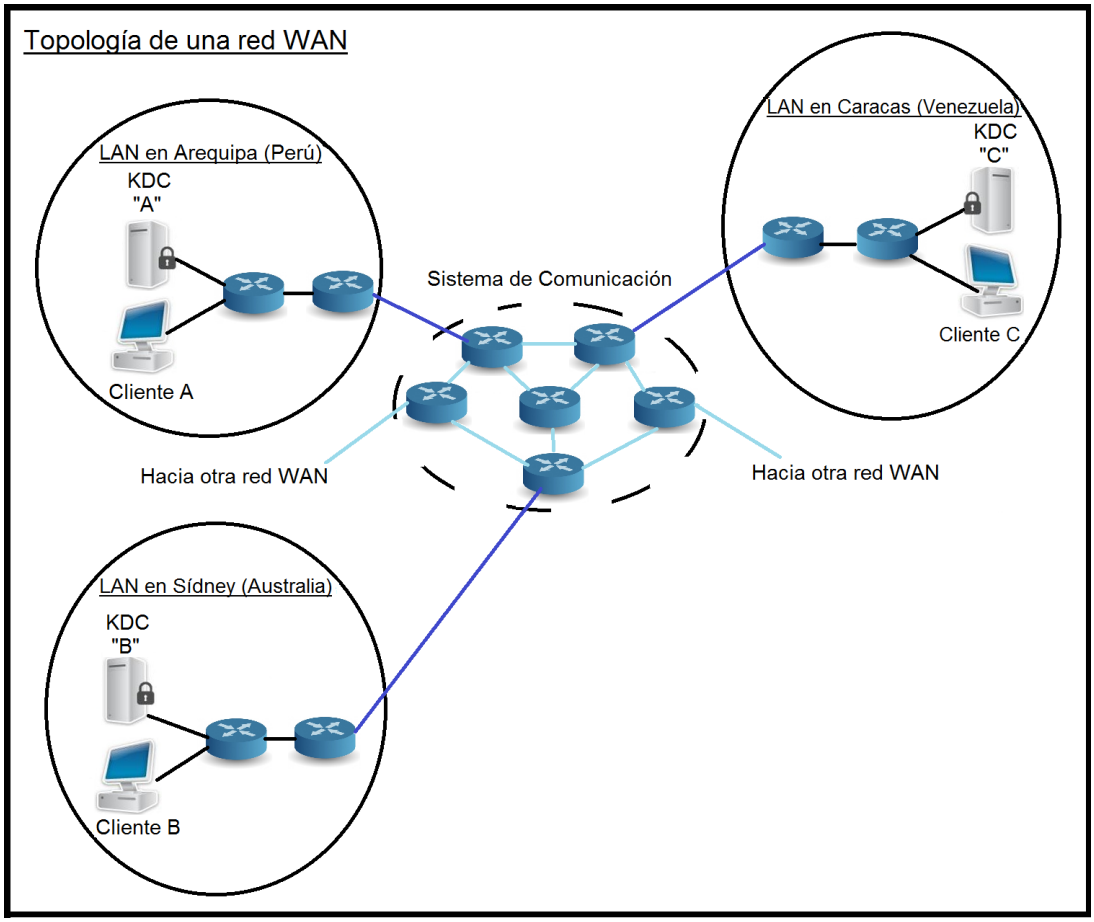

Figura 4.10: Topología de una red WAN.

La forma en la que esta propuesta actuará en una red WAN no es muy diferente de la topología en la red LAN, y habiendo explicado la forma en que opera en el escenario anterior, para este segundo escenario resultara más fácil y más rápido.

Como observará en la Figura 4.10 existen 3 KDCs diferentes en cada una de las redes, estos servidores KDCs estarán en constante comunicación entre ellos, a través de los routers en el medio.

Al plantear la situación en la que el cliente "A" desea comunicarse con el cliente "B", y el cliente "B", con el "C", tendremos el siguiente proceso de funcionamiento de la propuesta "IPv6 Kerberos".

A. Registro de los clientes en la red WAN

Primero y como se observa en la Figura 4.11, el proceso de registro entre los 3 clientes es el mismo. Es decir, que los 3 clientes enviarán sus mensajes de registros, con las peticiones de que desean comunicarse, a sus respectivos KDCs y estos KDCs se comunicaran con los otros KDCs con los otros KDCs, una vez que hayan verificado la identidad de sus respectivos usuarios.

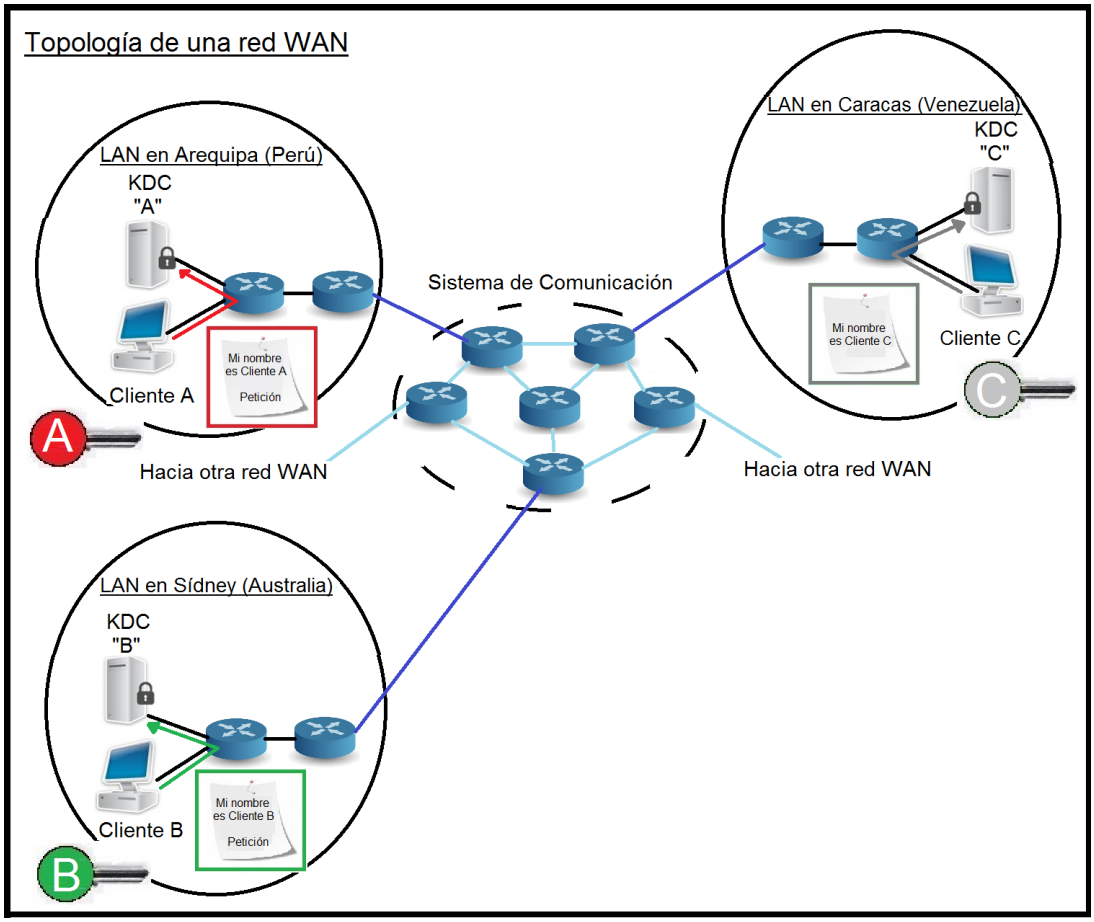

Figura 4.11: Registro de los clientes en la red WAN.

B. Comunicación de los KDCs red WAN

Como se observa en la Figura 4.12, esta comunicación entre KDCs, se realiza dado que si el cliente “A” desea comunicarse con “B”, entonces debe recibir el ticket de “B”, sin embargo el KDC de “A”, no posee dicho ticket, por ello el KDC “A”, se comunica con el KDC “B”, para solicitar dicho ticket y así entregárselo al cliente “A”, el mismo proceso se repetirá para los clientes “B” y “C”.

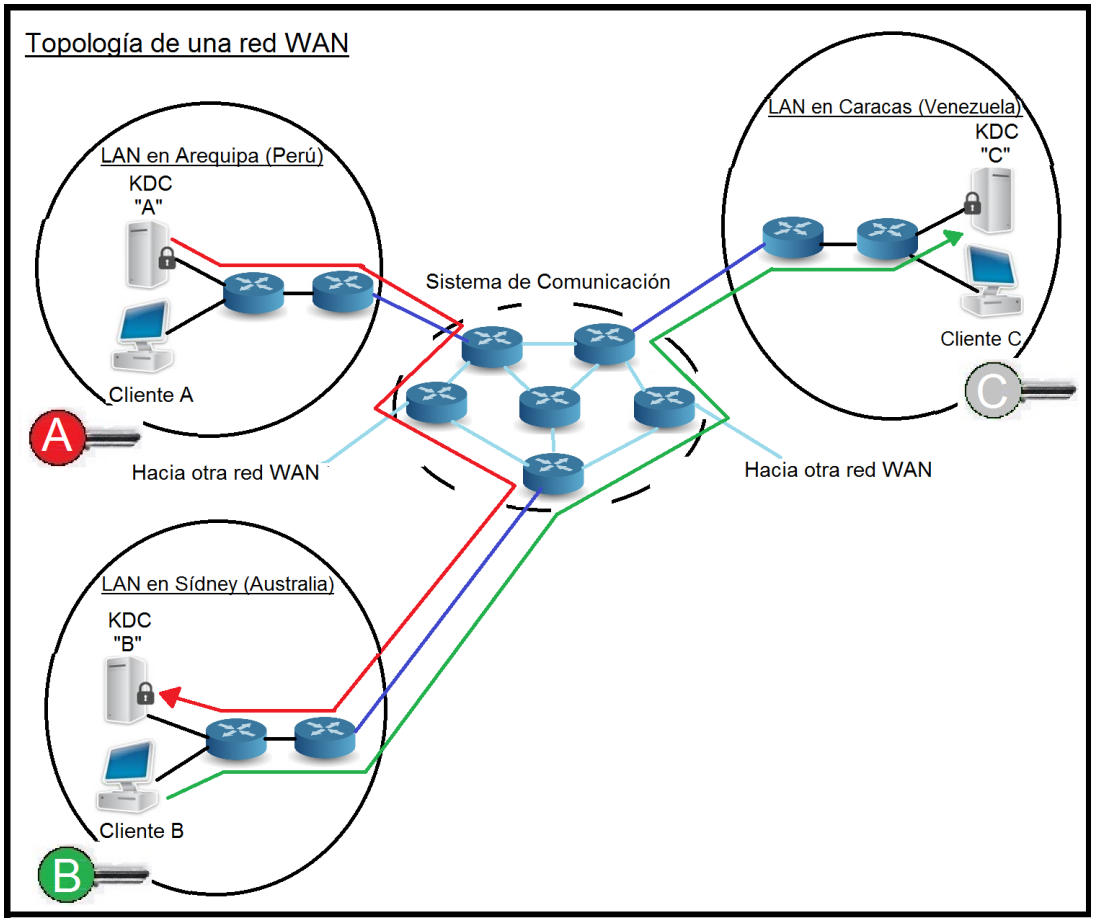

Figura 4.12: Comunicación de los KDCs red WAN.

C. Obtención de los tickets de los KDCs red WAN

Como se observa en la Figura 4.13, una vez que los KDCs se comuniquen con los KDCs respectivos, y de acuerdo a la petición de los clientes con quien desean comunicarse, solicitaran los TGTs necesarios y recibirán los TGTs para enviárselos a los clientes, bajo la misma seguridad de llaves que en la topología LAN.

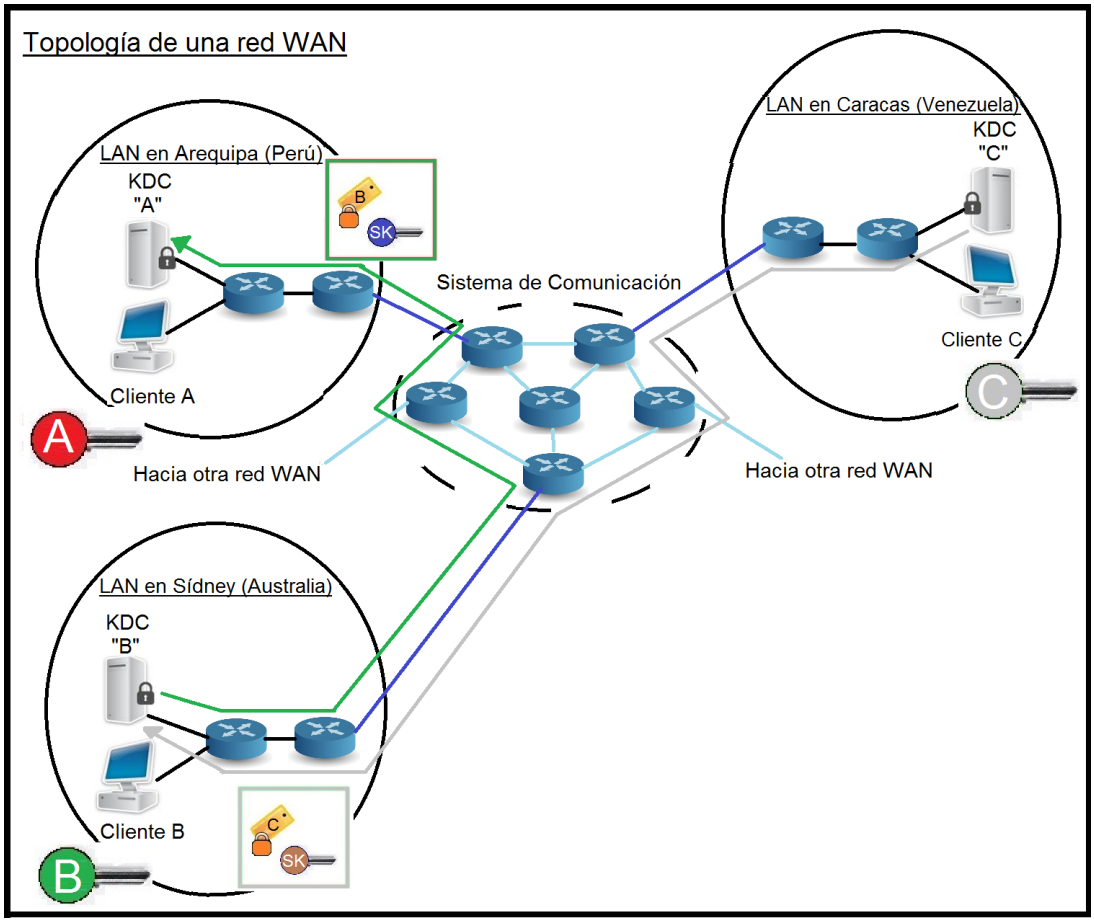

Figura 4.13: Obtención de los tickets de los KDCs red WAN.

D. Obtención de los tickets de servicio de los KDCs red WAN

A continuación y como se observa en la Figura 4.14, los clientes enviarán autenticadores destinados a los KDCs de los que recibieron los tickets, y estos KDCs realizaran las verificaciones respectivas, para luego enviarles los tickets de servicios que necesiten.

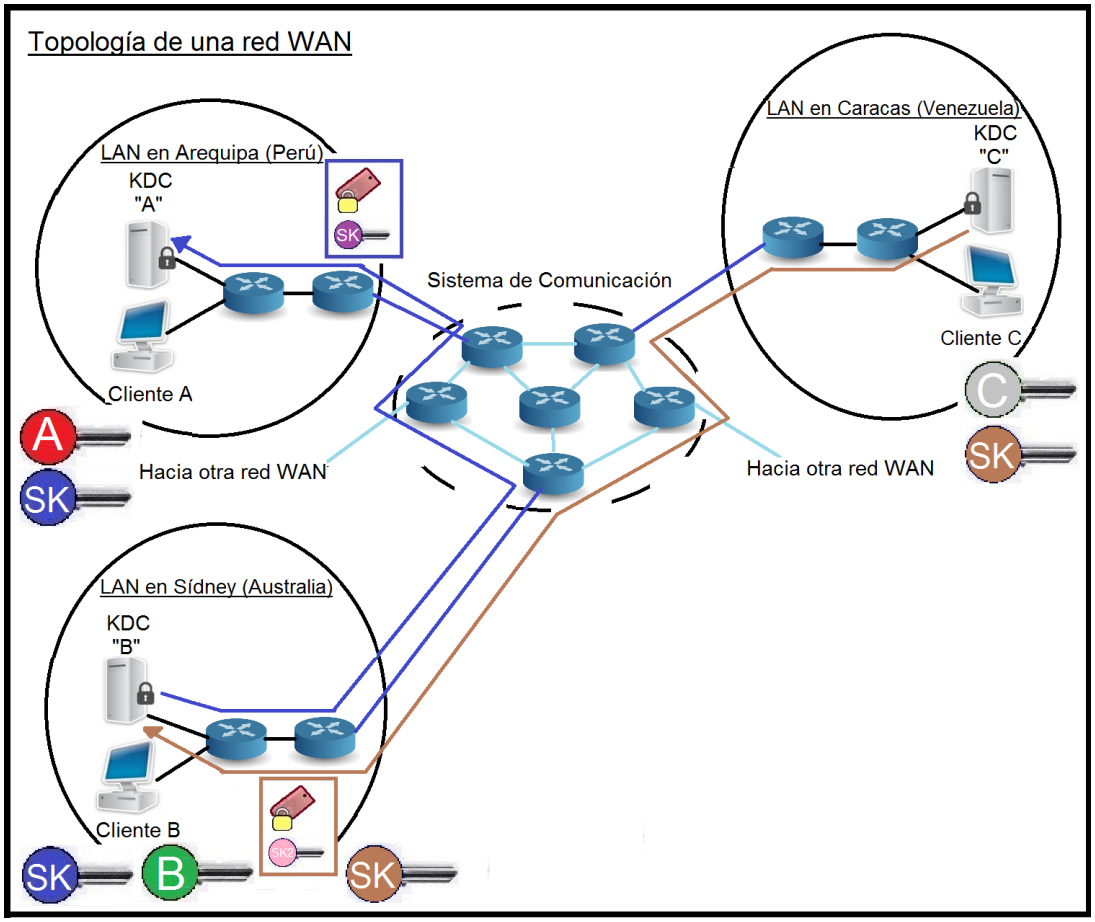

Figura 4.14: Obtención de los tickets de servicio de los KDCs red WAN.

E. Obtención de las llaves privadas de los KDCs red WAN

Además de los tickets de servicio, y como se observa en la Figura 4.15, a aquellos clientes que no envíen peticiones (Cliente "B" para su comunicación con "A" y Cliente "C" para su comunicación con "B"), se les enviara las respectivas copias de las llaves, con las que podrán abrir los tickets de servicio que reciban, para así realizar sus corroboraciones respectivas con los autenticadores que recibirán luego.

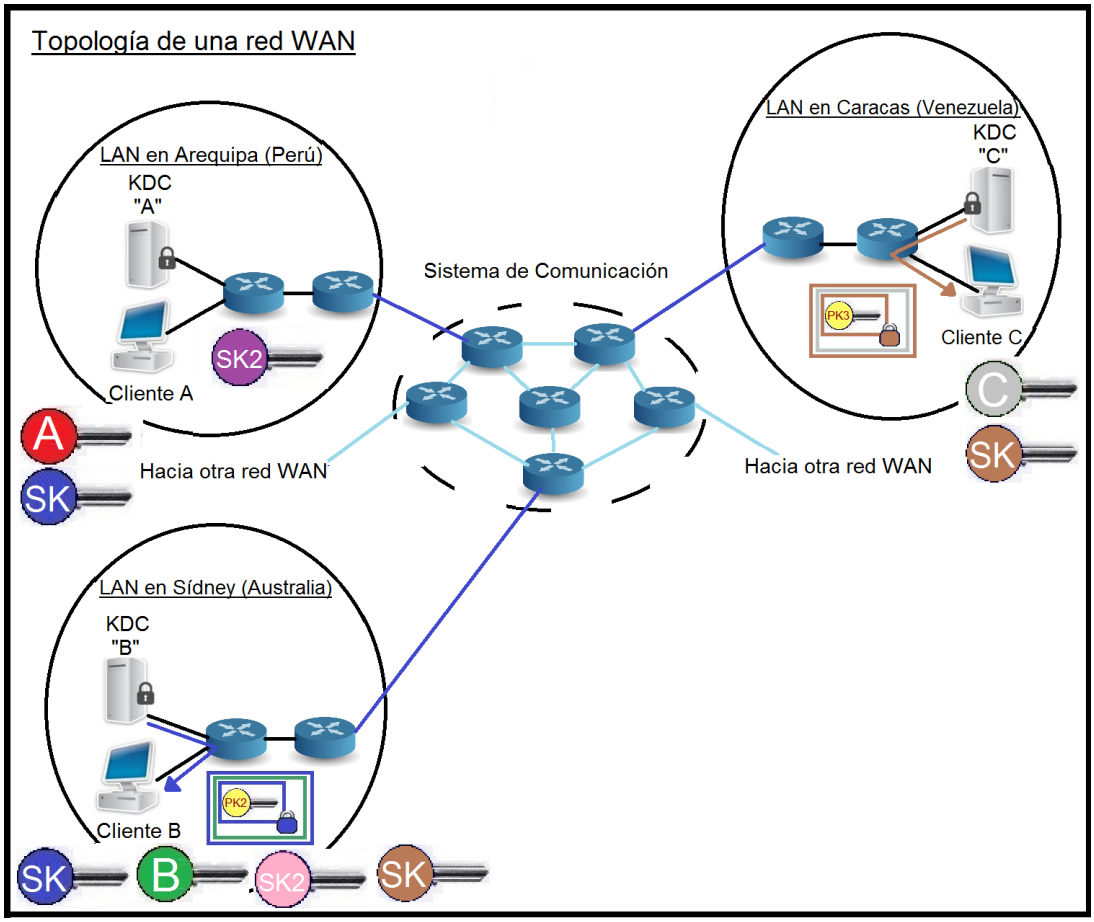

Figura 4.15: Obtención de las llaves privadas de los KDCs red WAN.

F. Envío de los paquetes IPv6 de los clientes con la solicitud red WAN

Finalmente y como se observa en la Figura 4.16, se enviaran los paquetes IPv6 respectivos de los clientes, con los autenticadores y los tickets de servicio, para que los clientes receptores puedan corroborar la identidad de los emisores y así se establecerá las comunicaciones necesarias.

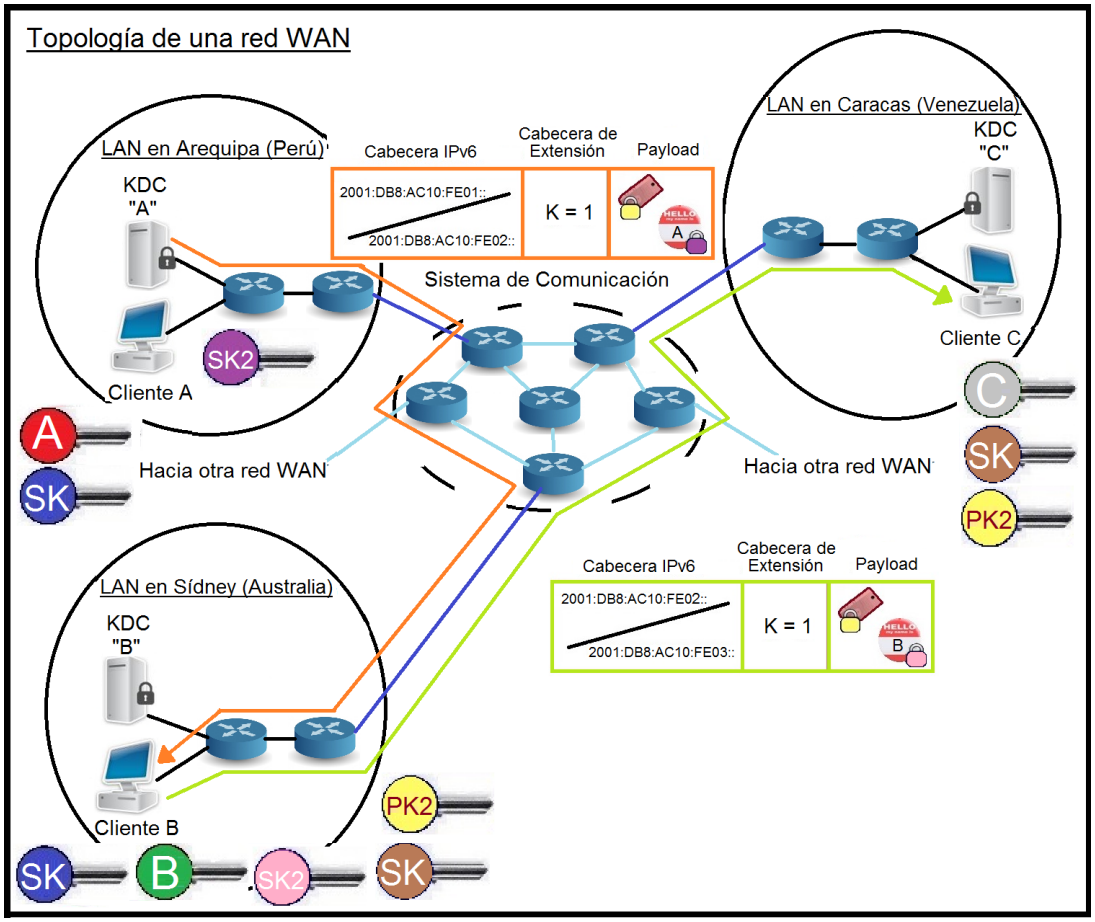

Figura 4.16: Envío de los paquetes IPv6 de los clientes con la solicitud red WAN.

Lo último que queda hacer es enviar el mensaje IPv6 de retorno, con la respuesta y solicitud resuelta, de esta forma se realiza la comunicación en una red WAN. Como podrá notar el proceso es el mismo que en una red LAN, con la diferencia del tamaño de la red y que existe una comunicación entre los KDCs, además el único cambio en la información al enviarse y cambiar de red, es solo la trama, más los datos no cambian.

4.3. Comunicación entre clientes IPv6, con canal IPv4

En esta tercera topología, se presenta el escenario de dos clientes que se desean comunicar, de la misma forma que en la topología de la red LAN, con la diferencia que en el medio existen equipos IPv4, por lo que se debe existir una forma de permitir el envió de los mensajes IPv6, pasando por IPv4.

Mencionamos al inicio de la presentación que con la migración a IPv6, los equipos IPv4 se eliminarán, sin embargo dado que aún no se ha dado la total migración, para un escenario como el tercero, no se puede esperar a la migración, y por ello se debe realizar la comunicación entre IP4 e IPv6.

Para empezar mostraremos en la Figura 4.17 la topología de este tercer escenario, en el cual se observa una red WAN, donde existen dos clientes IPv6 con sus 2 KDCs, y demás equipos en IPv6, sin embargo en el canal se estará operando con IPv4.

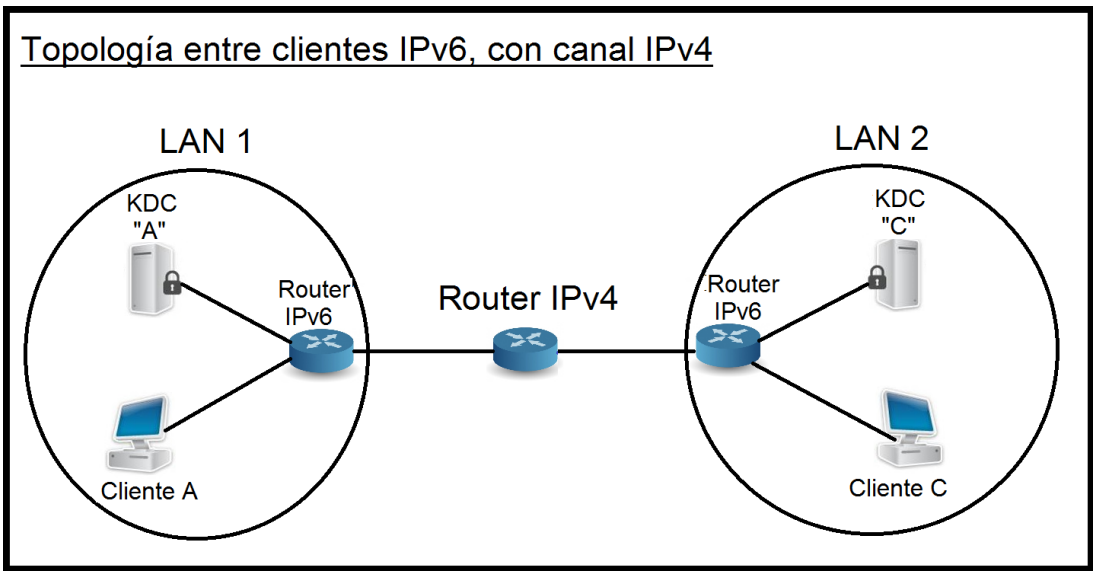

Figura 4.17: Topología de dos clientes IPv6 con un canal IPv4.

A. Envío del paquete IPv6, del cliente A, al cliente C.

Para la siguiente topología, su procedimiento no difiere de la redWAN en la sección anterior. Existe una comunicación entre los 2 KDCs y una negociación en el medio, él envió de las llaves en los mensajes y paquetes será el mismo. La única diferencia radica al final de la operación, en el momento en que se envían los paquetes IPv6.

La diferencia solo surgirá con el router IPv4, el cual no alterara los datos del paquete IPv6 que reciba. El router IPv4, recibirá el paquete IPv6, observara la dirección del emisor y destino, y enrutará dicho mensaje IPv6 en un paquete IPv4, donde en su Payload ingresara todo el paquete IPv6, para enviarlo a su destino.

Mientras que en lado del receptor, se recibirá el paquete IPv4 y se accederá al paquete IPv6, de manera que los datos no han sido alterados, más la trama si se alteró y el router IPv4, solo cumplió la tarea de recibir el paquete IPv6, y dejarlo pasar en

un paquete IPv4.

El router IPv4 contara con la capacidad para poder recibir e interpretar el paquete IPv6 que reciba y podrá enrutar todo ese paquete en un nuevo paquete IPv4, mientras que en lado del receptor, su router o routers IPv6 contaran con la capacidad para poder recibir e interpretar el paquete IPv4 y des-entramarlo, para acceder a la información.

Para un mejor entendimiento, se puede observar la Figura 4.18, en la que se muestra el proceso final del envío del paquete IPv6, desde el cliente "A", al cliente "C".

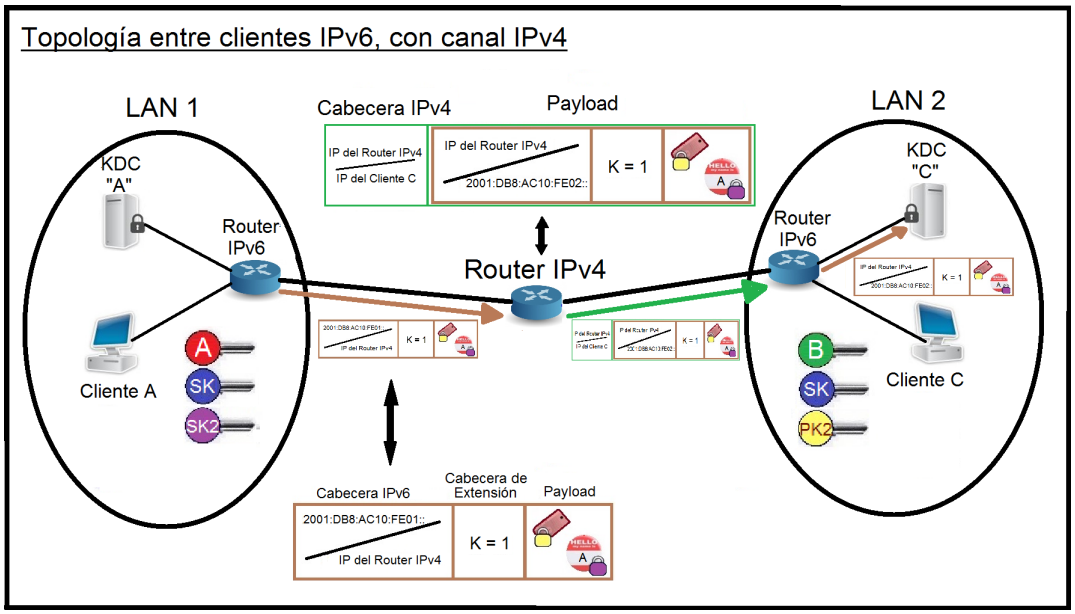

Figura 4.18: Envío del paquete IPv6, del cliente A, al cliente C.

Finalmente se repetirá la misma operación, sin embargo de forma inversa, desde el cliente C, al cliente A, para enviarle la confirmación y la resolución de la petición, y de esta manera se resuelve esta topología de IPv6 Kerberos, con IPv4.

5 Implementación, simulaciones y resultados

En esta sección se detalla la parte de la implementación y simulación de la nueva propuesta. Como se mencionó con anterioridad el lenguaje de programación que se optó por utilizar, es C en la plataforma de Linux. Para ello se creó una máquina virtual con Ubuntu, y se realizó una comunicación entre Terminales en modo Root de Linux.

El primer paso para la implementación de la nueva propuesta, fue crear la unidad de transmisión de mensajes IP, el cual llevara un socket con el mensaje en la carga o payload del mensaje. El fundamento de enviar un socket con un mensaje en la carga se aplicara en todo momento para enviar el mensaje de registro inicial al KDC, el envió del ticket de sesión y de servicio, el envió del autenticador y finalmente el envió del mensaje con la solicitud y las respuestas o confirmaciones que se presentaran en todo momento.

Una vez que se logre enviar un mensaje desde un terminal a otro, el segundo paso es crear una unidad de recepción, la cual cumplirá con las siguientes tareas:

- Recibir el mensaje (socket) que se envió con la unidad de transmisión.
- Mostrar el mensaje tal cual se envió.
- Procesar el mensaje recibido.

El tercer pasó es llevar la comunicación a la forma en que el protocolo Kerberos opera, teniendo las unidades de transmisión y recepción se implementa y simula una comunicación en Kerberos. Es decir que la comunicación pasara de 2 terminales, a operar entre 3 terminales. Debido a que Kerberos realiza un proceso de autenticación entre el KDC y un cliente Kerberos, para luego realizar él envió de tickets, se utilizaran en varias ocasiones las unidades de transmisión y recepción anteriormente implementadas.

Finalmente se presentan los resultados obtenidos de la nueva propuesta, y se la compara con una comunicación IP que no utilice Kerberos. Se analiza los resultados de las comparaciones, se presentan los resultados y se indican las conclusiones finales.

5.1. Diagrama de la comunicación propuesta

Hemos mencionado que una vez implementadas las unidades de transmisión y recepción, repetiremos su uso durante toda la comunicación entre los 3 terminales creados. Si observa nuevamente el apartado 4 del informe observara que en la comunicación planteada por la propuesta, en todo momento se transmiten mensajes, ya sea de los clientes al KDC, o del KDC a los clientes, o entre clientes Kerberos. En todo momento se transmite información entre los diferentes puntos y la información enviada será diferente para cada tramo. Y de la misma forma que se envía información, también se recepciona información, la cual también será diferente en cada tramo y requerirá de un procesamiento especifico.

Si observamos la operatividad de la propuesta en el punto 4, viéndola desde un punto de vista de emisor y receptor, cada uno de los 3 terminales se convertirá en emisor y receptor en todo momento. De forma específica antes de que se realice la comunicación entre clientes, existirán 4 comunicaciones previas entre el KDC y un cliente Kerberos, 1 primera comunicación de identificación entre los clientes, y el intercambio de mensajes entre los clientes, el cual como mínimo puede ser 2 comunicaciones.

En resumen estamos hablando de 7 trayectos de comunicaciones, 7 trayectos donde se utilizaran 7 veces la unidad de transmisión, y 7 veces la unidad de recepción. En la Figura 5.1, se puede observar el diagrama con las 7 comunicaciones de la propuesta planteada, además del contenido del mensaje que se envía en cada tramo.

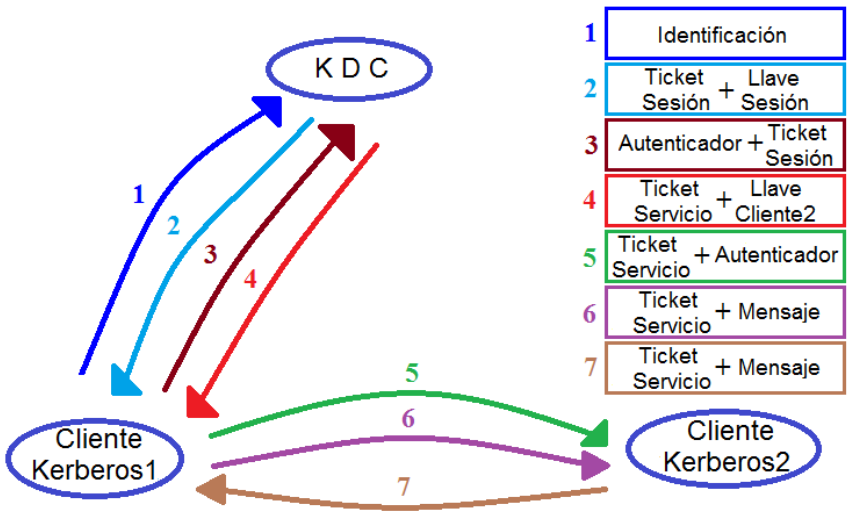

Figura 5.1: Diagrama de la comunicación propuesta.

5.2. Unidad de transmisión y recepción

El código de la unidad de transmisión de paquetes, se encuentra anexado al final del informe. Esta unidad de transmisión envía un mensaje UDP, con ayuda de un socket. En este mensaje se ingresa los Datos, se ingresa la dirección IP, el número del puerto que se utilizara y se envía el mensaje, con la línea de comando sendto(). La unidad de transmisión, le pide al usuario ingresar el mensaje que desea enviar al receptor, dicho mensaje ingresado por teclado se guardara en una variable y se utiliza el comando sendto() con dicha variable para poder ser enviado.

El código de la unidad de recepción, también se encuentra anexado al final del informe. Esta unidad crea un socket e ingresa sus parámetros con información del socket que se envió, tal y como el mismo número de puerto, es decir que se creara un socket donde se guardara la información que se recibirá. Con ayuda del comando recvfrom () se recibirá el mensaje enviado, guardándolo en una nueva variable y para efectos de prueba se imprimirá el mensaje.

Para demostrar que el mensaje se envió y llego a su destino, abriremos 2 terminales de comunicación e ingresaremos al modo "root" en ambos terminales, tal y como se observa en la Figura 5.2.

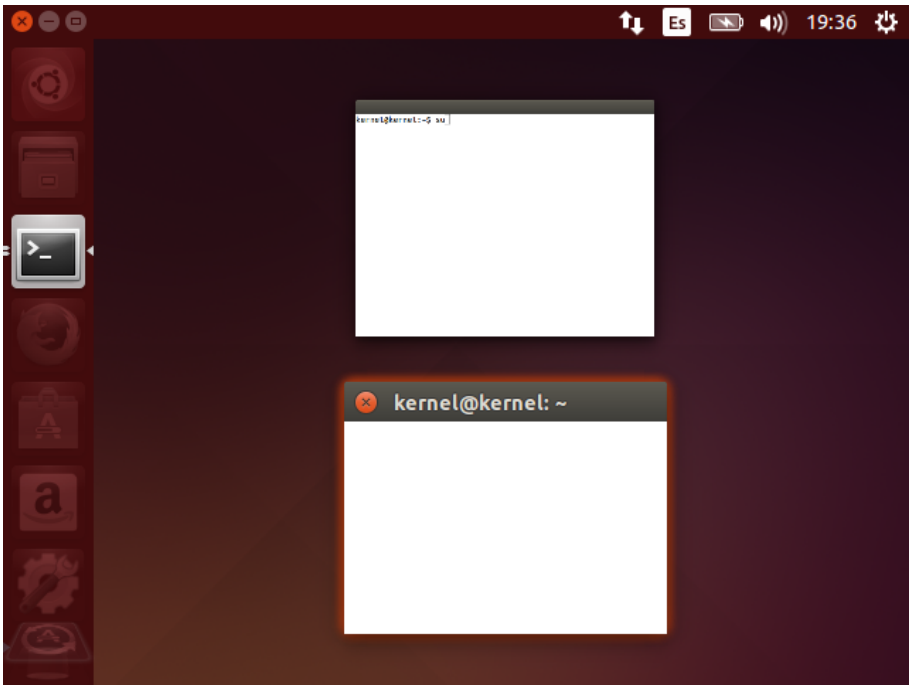

Figura 5.2: Terminales de comunicación.

El siguiente paso es leer y ejecutar la unidad de transmisión y recepción, tal y como se observa en las Figuras 5.3 y 5.4, cada uno en uno de los terminales, así el terminal que ejecute la unidad de transmisión, actuara como un cliente, mientras que el segundo terminal, al ejecutar la unidad de recepción, actuara como un servidor.

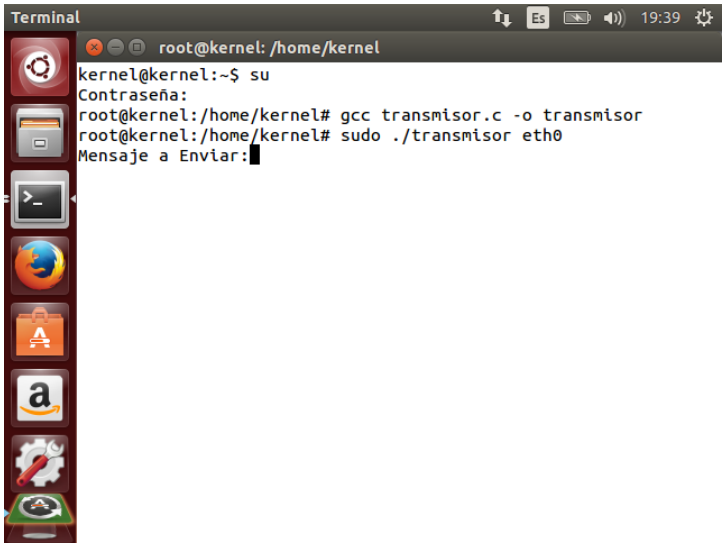

Figura 5.3: Terminal de emisión en modo root ejecutando la unidad de transmisión.

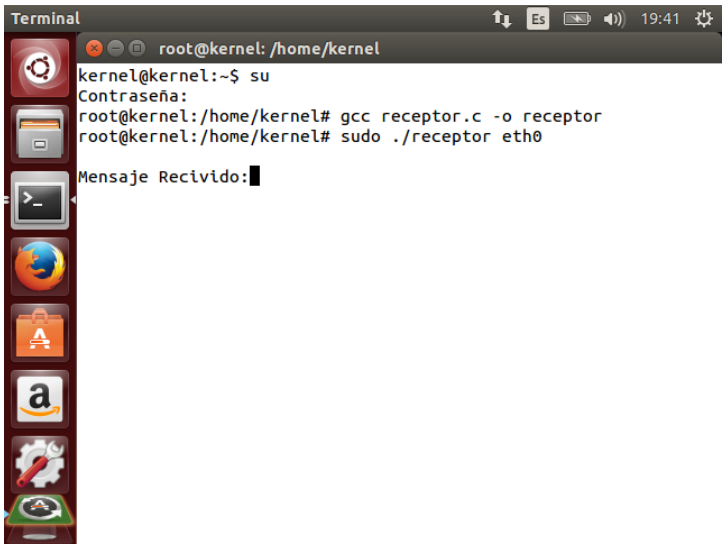

Figura 5.4: Terminal de recepción en modo root ejecutando la unidad de recepción.

Finalmente ingresamos un mensaje de prueba por teclado desde el terminal de emi-

sión, y observamos los resultados en el terminal de recepción, tal y como se observan en las Figuras 5.5 y 5.6.

Figura 5.5: Mensaje de prueba a enviar

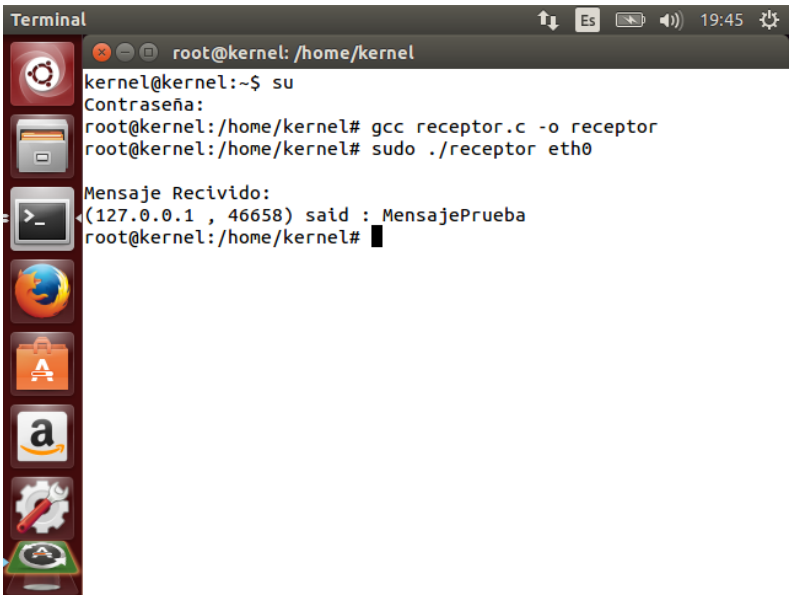

Figura 5.6: Resultados de la primera linea de comando en el terminal de destino.

5.3. Comunicación entre los 3 terminales

En este punto implementaremos la nueva propuesta, haciendo uso de las dos unidades creadas, para lograr una comunicación IPv6 con autenticación Kerberos. Para ello se ha optado por crear 3 nuevas unidades, cuyos códigos se encuentrán anexados al final del informe.

La primera unidad será el primer cliente Kerberos, el cual llamaremos "cliente 1", esta unidad comprenderá todas las actividades de transmisión de información y recepción de información que realizará un cliente Kerberos que desea comunicarse con otro cliente Kerberos. Todas las funciones que realizara este cliente se pueden observar en la Figura 5.1, expresadas como las operaciones del "cliente Kerberos 1".

De forma precisa la unidad "cliente 1" durante todo su proceso para la propuesta implementada, realizara 7 tareas específicas, 4 transmisiones de datos y 3 recepciones de mensajes. Además de transmitir los mensajes, también deberá encriptar dichos mensajes antes de ser enviado, con las respectivas llaves que utilizara.

Hemos mencionado en un principio que un cliente Kerberos utilizara, en un principio, su propia llave y luego utilizara una copia de llave de sesión para encriptar sus mensajes. Por lo que también existe secciones de encriptación de los mensajes, tal y como se observa en los códigos anexados.

Y de la misma forma que se encriptan mensajes, para las recepciones especificas se des-encriptarán los mensajes recibidos para poder analizar sus contenidos, con la excepción de los tickets los cuales nunca podrán ser des-encriptados por ningún cliente Kerberos.

La segunda unidad será el "KDC", el cual realizara 4 tareas específicas, 2 recepciones de datos y 2 transmisiones de datos. Y de forma similar deberá realizar tareas de des-encriptación y encriptación para todos los mensajes.

La última unidad implementada será el segundo cliente Kerberos, el cual llamaremos "cliente 2", esta unidad comprenderá 3 tareas específicas, 2 recepciones y 1 transmisión de datos, además de las des-encriptaciones y encriptaciones específicas.

Finalmente la comunicación IPv6 se observa en la última transmisión y recepción del "cliente 1", con la última recepción y la única transmisión del "cliente 2". Todo el proceso anterior es netamente autenticación en Kerberos.

Empezaremos por abrir 3 terminales, tal y como se observa en las Figuras 5.7 y 5.8, e ingresamos a modo “root” en cada terminal de comunicación, para ejecutar los respectivos códigos.

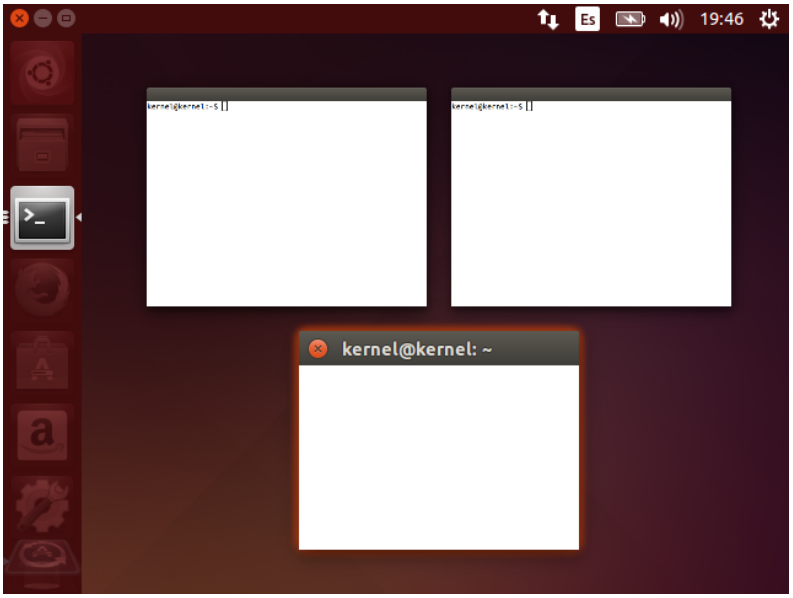

Figura 5.7: Terminales de comunicación para la nueva propuesta.

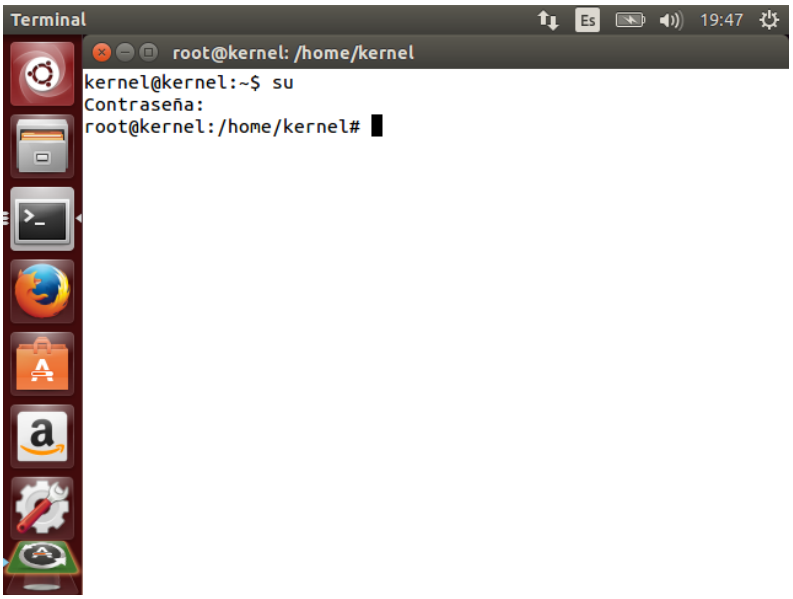

Figura 5.8: Terminales de comunicación para la nueva propuesta en modo root.

A continuación leeremos y ejecutaremos la unidad “cliente 1” en un terminal, y la unidad “KDC” en otro terminal, tal y como se aprecia en las Figuras 5.9 y 5.10. El tercer terminal se utilizara para el cliente 2, pero todavía no ejecutaremos el código “cliente 2”.

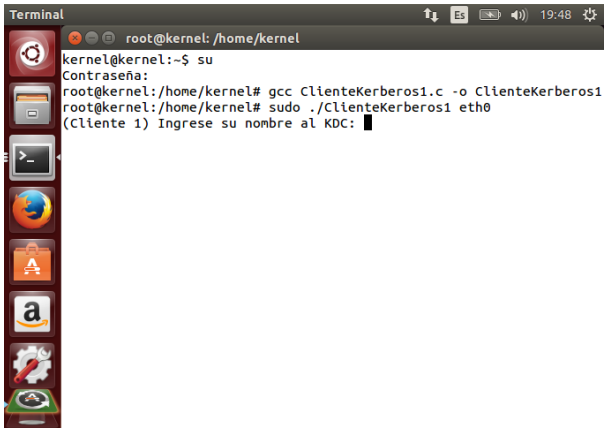

Figura 5.9: Cliente Kerberos iniciando la comunicación con el KDC.

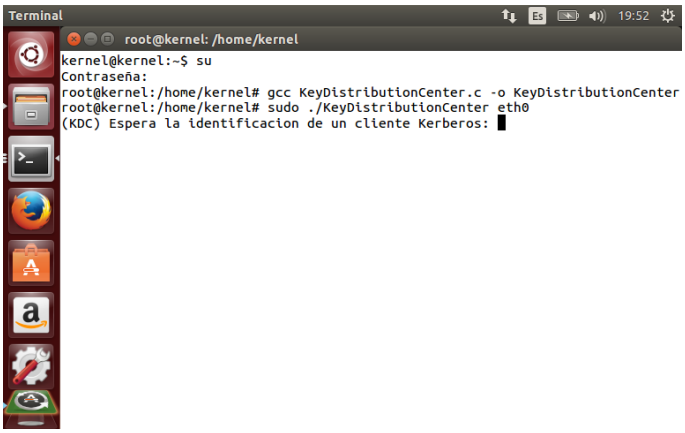

Figura 5.10: KDC esperando la identificación de un cliente Kerberos.

El siguiente paso es ingresar el identificador del cliente 1 al KDC, hemos mencionado que en un sistema Kerberos, todos trabajan con Kerberos, por lo que el KDC tiene una base de datos con el nombre del cliente 1, así como su IP, su llave privada y otros parámetros. De esta forma el KDC detectara en primera instancia la identidad del cliente 1, corroborando su nombre, la IP del mensaje que recibido y otros parámetros. Los resultados se observaran en las Figuras 5.11 y 5.12.

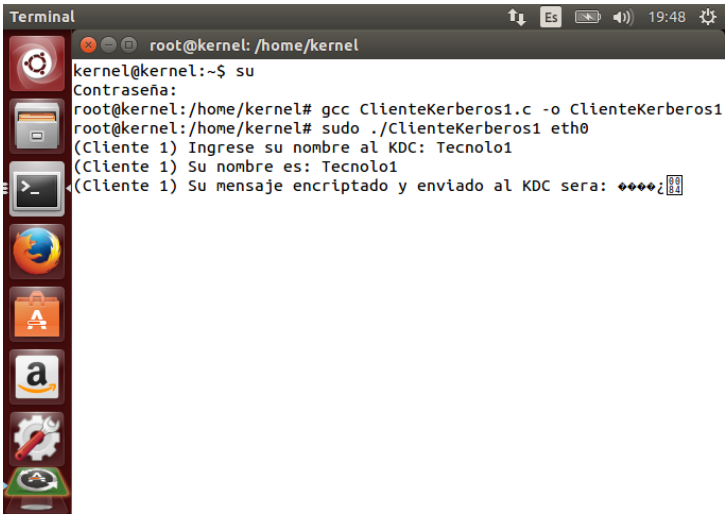

Figura 5.11: Cliente Kerberos identificandose al KDC.

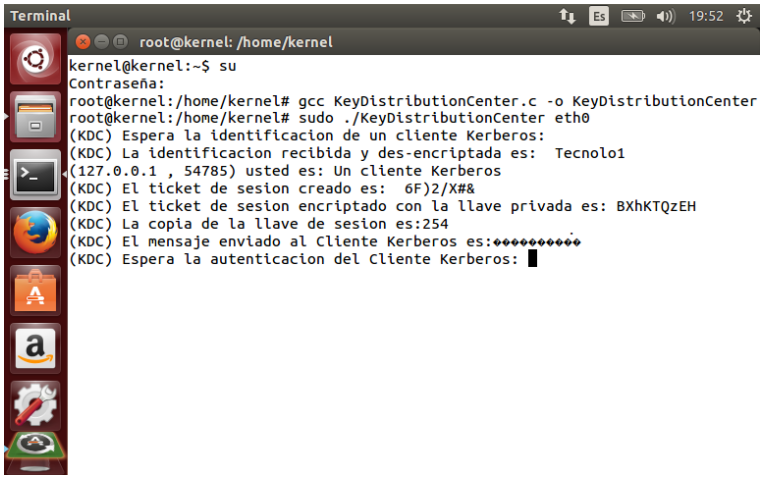

Figura 5.12: KDC aprobando la identidad del cliente Kerberos y enviandole un ticket con una nueva llave.

Como observamos en la Figura 5.12, el KDC ha aprobado la identidad del cliente 1 y ha creado un ticket de sesión además de una copia de la llave de sesión y le ha enviado dicha información al cliente 1, protegiéndola con la llave privada del cliente 1. En este punto el KDC espera que el cliente 1 se vuelva a identificar mandándole un autenticador junto con el ticket que le envió. En la Figura 5.13 vemos como el mensaje enviado por el KDC, fue recibido por el cliente 1. Ahora el cliente 1 debe enviar su autenticador al KDC, para recibir un ticket de servicio. Tal y como se observa en la Figura 5.14.

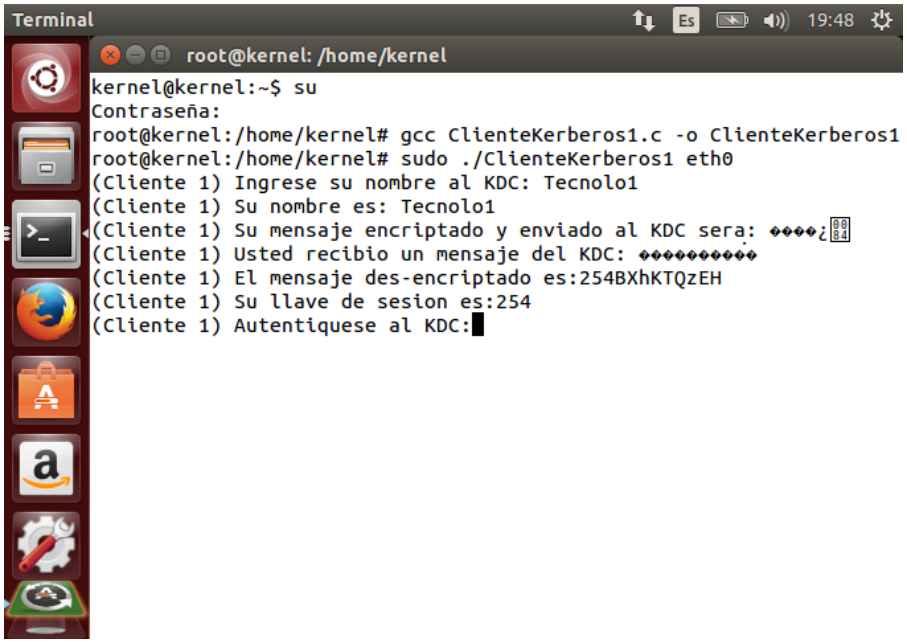

Figura 5.13: Cliente Kerberos recibe su ticket de sesión por el KDC.

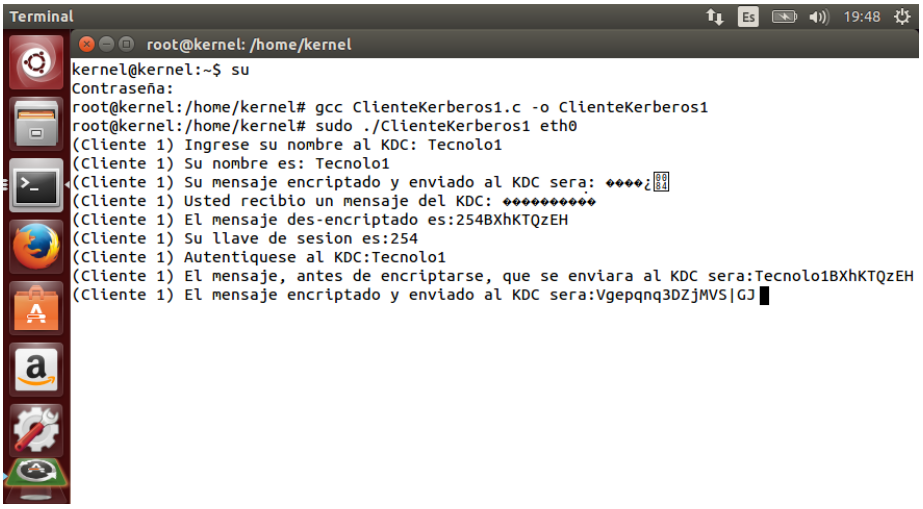

Figura 5.14: Cliente Kerberos mandando su autenticador al KDC.

Continuando con la comunicación, si observamos la Figura 5.15, veremos como el terminal del KDC recibió el autenticador del cliente 1, nuevamente después de procesar la información recibida, de ser positiva creara un ticket de servicio y enviara dicho ticket junto con la llave privada del cliente 2, para que los clientes se puedan comunicar. En la Figura 5.16 el cliente 1 ha recibido el mensaje del KDC.

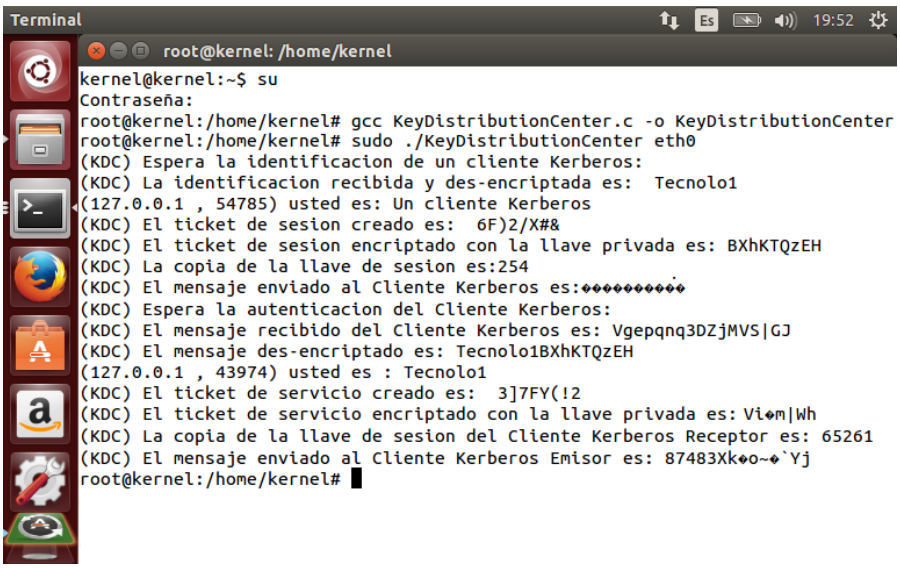

Figura 5.15: KDC procesa el autenticador y envia un ticket de servicio al cliente 1.

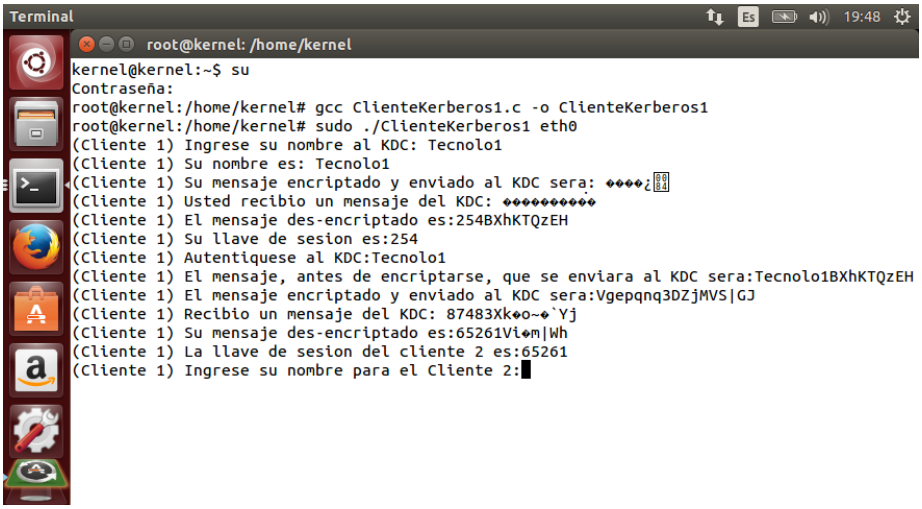

Figura 5.16: Cliente Kerberos recibe su ticket de servicio y la llave del cliente 2.

A partir de este punto las comunicaciones con el KDC han terminado, antes de identificarnos al cliente 2, necesitamos ejecutar en código del “cliente 2”, en el tercer terminal creado, tal y como se observa en la Figura 5.17 y luego nos identificamos al cliente 2, mandándole un mensaje encriptado con su propia la llave que conseguimos

del KDC, como se observa en la Figura 5.18.

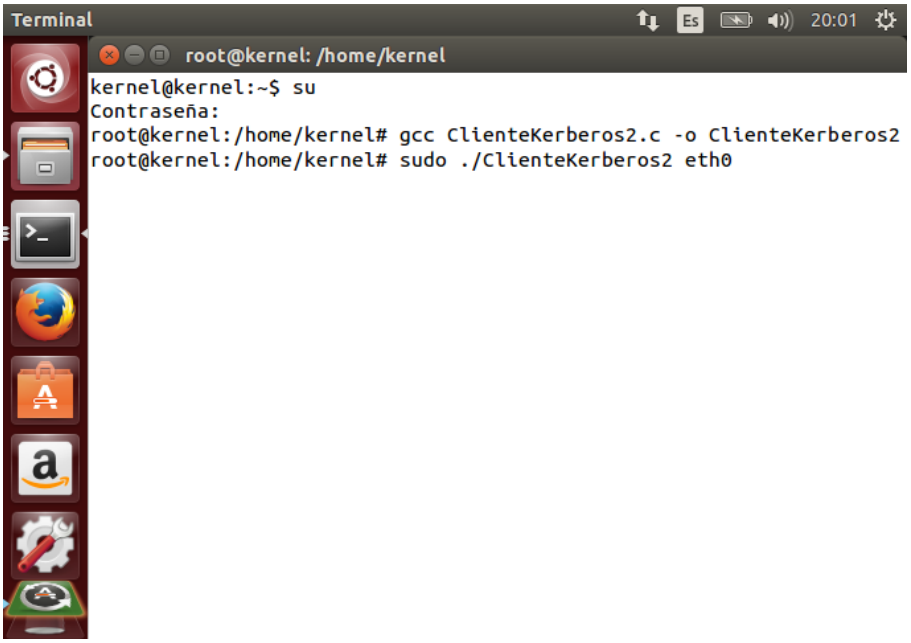

Figura 5.17: Cliente Kerberos 2 iniciando la comunicación con el cliente 1.

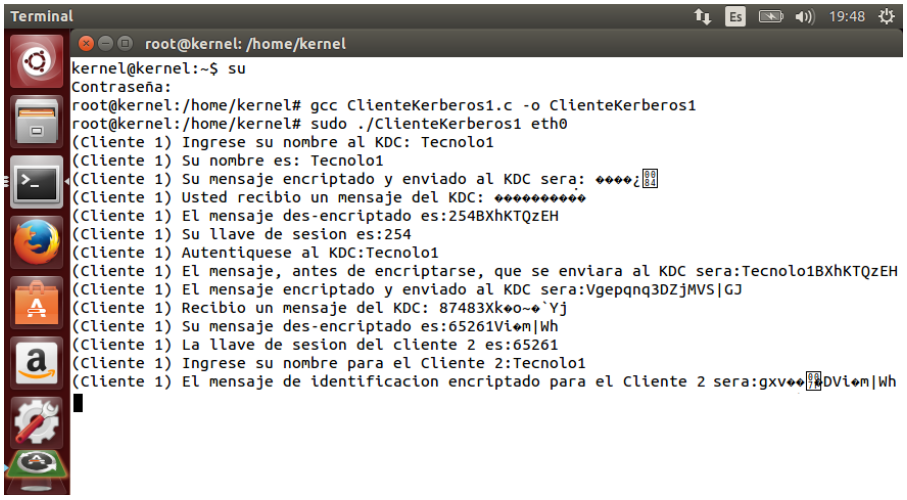

Figura 5.18: Cliente Kerberos 1 identificandose al cliente 2.

Observamos el terminal del cliente 2, para verificar si la identificación del cliente 1 llego al cliente 2, cabe mencionar que el cliente 2 podrá determinar la identidad del cliente 1, dado que nadie más tendrá su llave propia, ni el ticket de servicio con el

que cuenta. Como se observa en la Figura 5.19, de ser positiva la identidad, el cliente 1 recién podrá enviar su solicitud como se observa en la Figura 5.20.

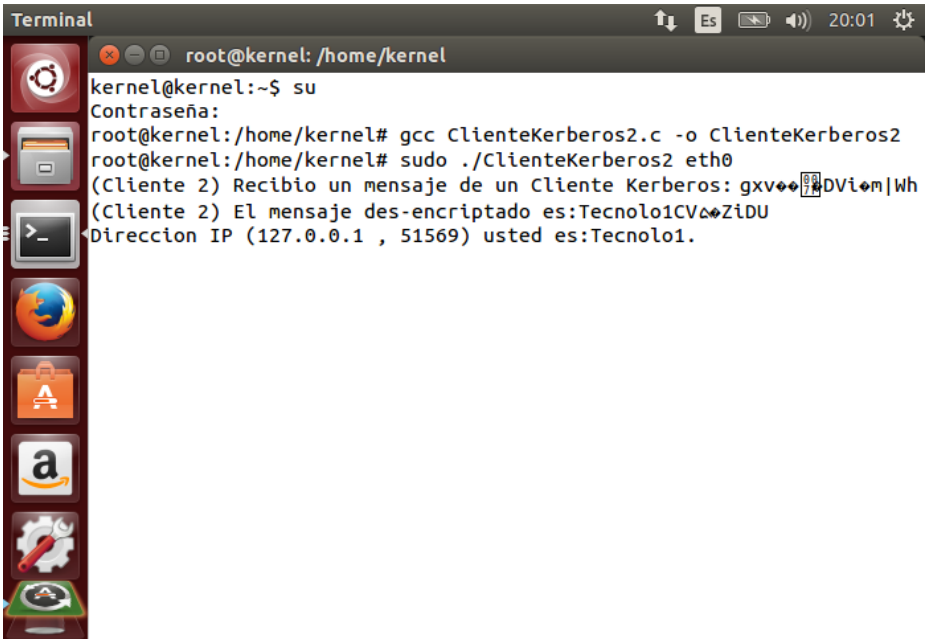

Figura 5.19: Cliente Kerberos 2 verifica la identidad del cliente 1.

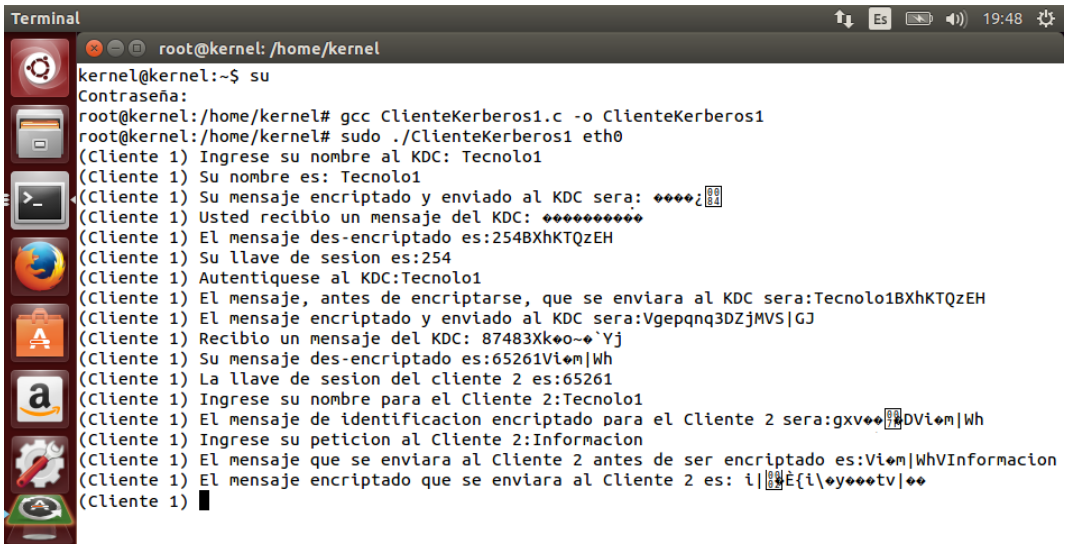

Figura 5.20: Cliente Kerberos envia una solicitud al cliente 2.

Una vez enviada la solicitud del cliente 1, al cliente 2, este último procesara la infor-

mación y leerá la petición del cliente 1, tal y como se ve en la Figura 5.21. Finalmente el cliente 2 resolverá la solicitud y enviara un mensaje al cliente 1 con la respuesta, tal y como se observa en la Figura 5.22.

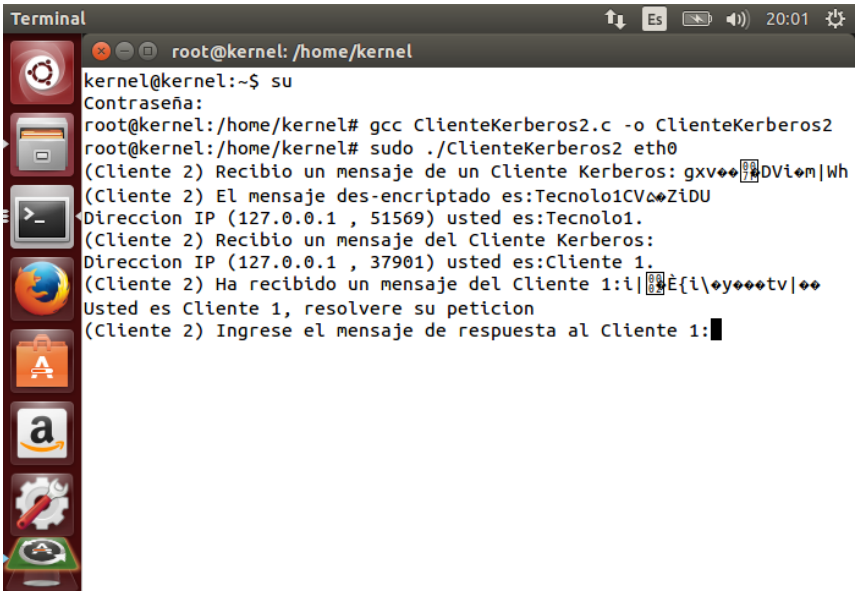

Figura 5.21: Cliente Kerberos 2 procesa y lee la solicitud el cliente 1.

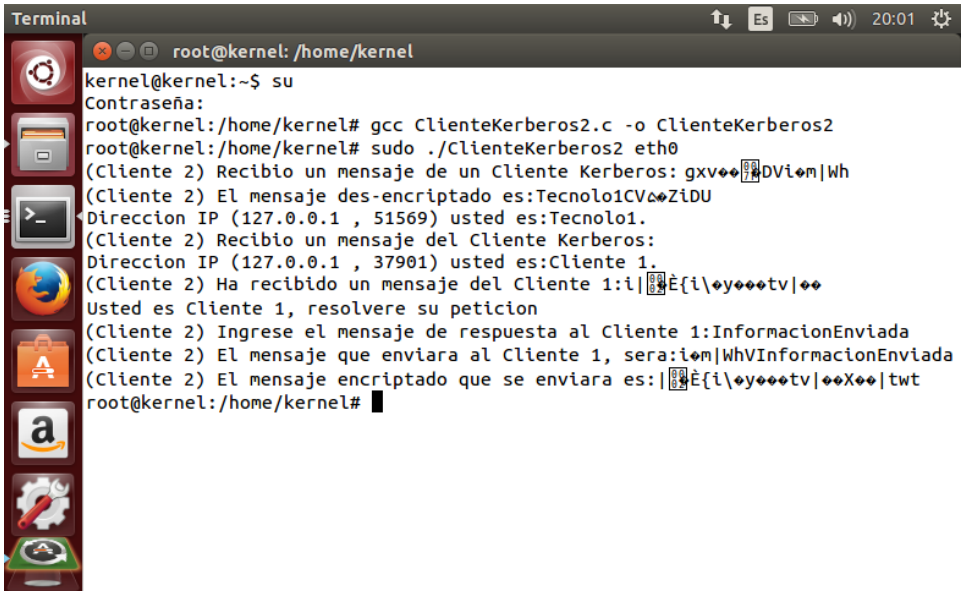

Figura 5.22: Cliente Kerberos 2 resuelve la solicitud del cliente 1.

Para concluir la comunicación el cliente 1 recibe el mensaje con la solicitud resulta por parte del cliente 2 y confirma que recibió la solicitud resuelta, como se observa en la Figura 5.23.

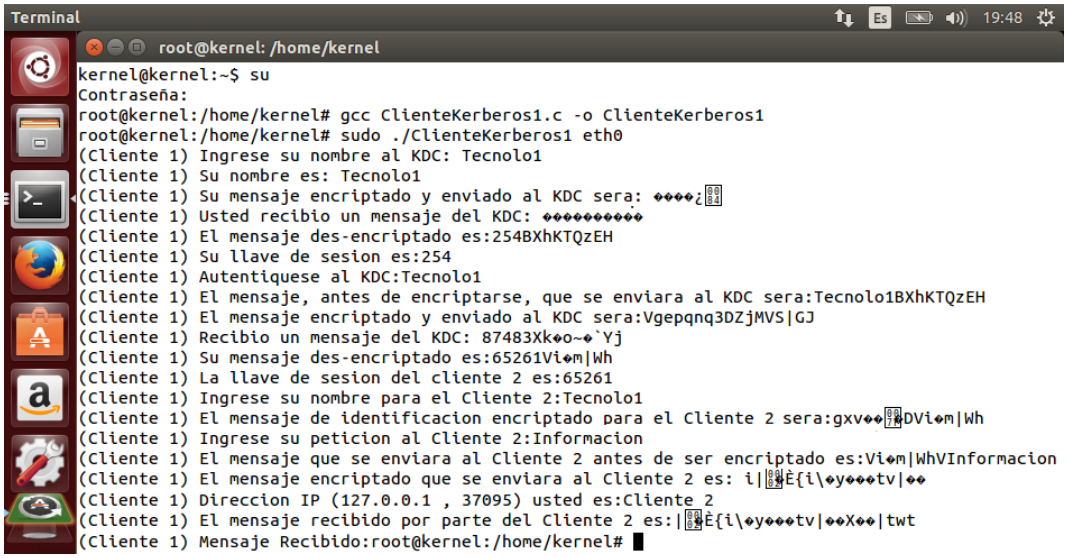

Figura 5.23: Cliente Kerberos 1 recibio la solicitud resuelta del cliente 2.

5.4. Comunicación entre los 3 terminales y un MITM

En esta sección probaremos la comunicación de la nueva propuesta con la presencia de un hombre en el medio, el cual intentara hacerse pasar por un cliente Kerberos. En las Figuras 5.24 y 5.25, observaremos el caso inicial donde un MITM intenta comunicarse con el KDC para recibir un ticket de sesión. Un MITM puede tener el mismo nombre del cliente Kerberos y hasta su llave, pero no podrá burlar la base de datos del KDC.

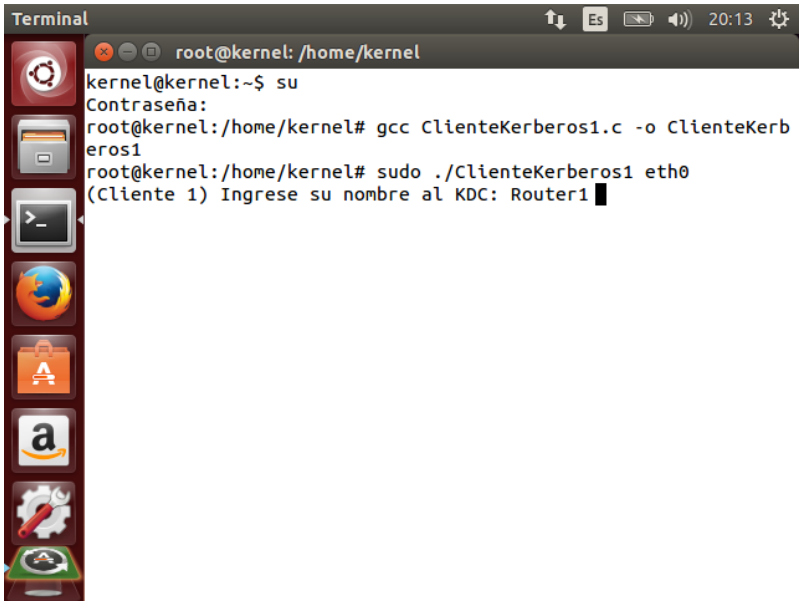

Figura 5.24: Hombre en el medio comunicandose con el KDC.

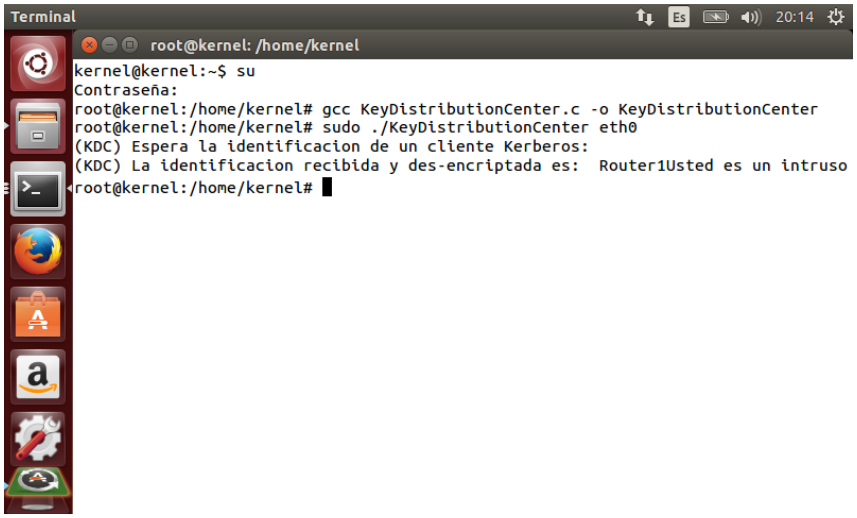

Figura 5.25: Respuesta del KDC al hombre en el medio.

El siguiente caso se aprecia cuando un MITM envía un autenticador al KDC para recibir en este caso un ticket de servicio, nuevamente el KDC analizara el mensaje enviado por el MITM. Al ingresar al autenticador del MITM y corroborar la información con contenido del ticket de sesión (al cual el KDC es el único que puede ingresar), lograra identificar al MITM y no procederá a continuar con la comunicación, tal y como se observan en las Figuras 5.26 y 5.27.

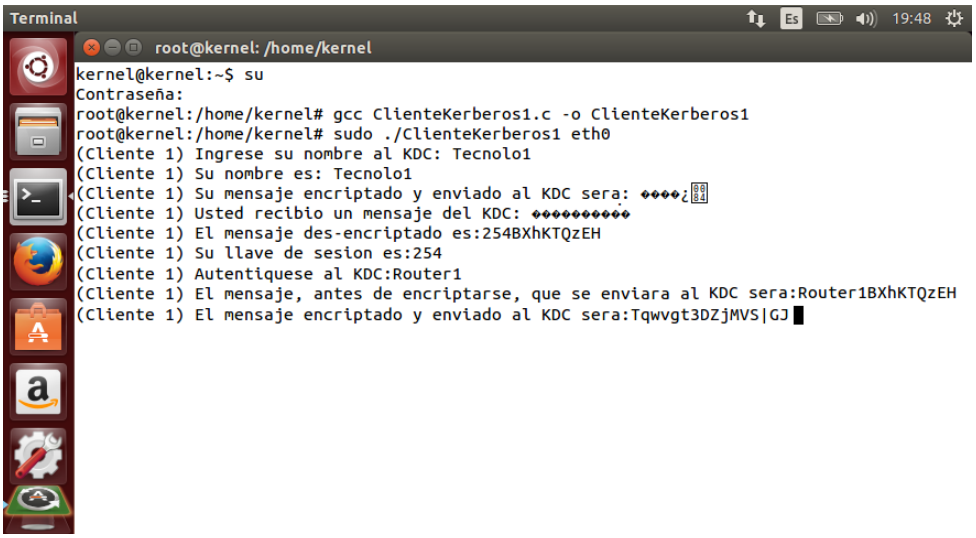

Figura 5.26: Hombre en el medio intentando conseguir un ticket de servicio del KDC.

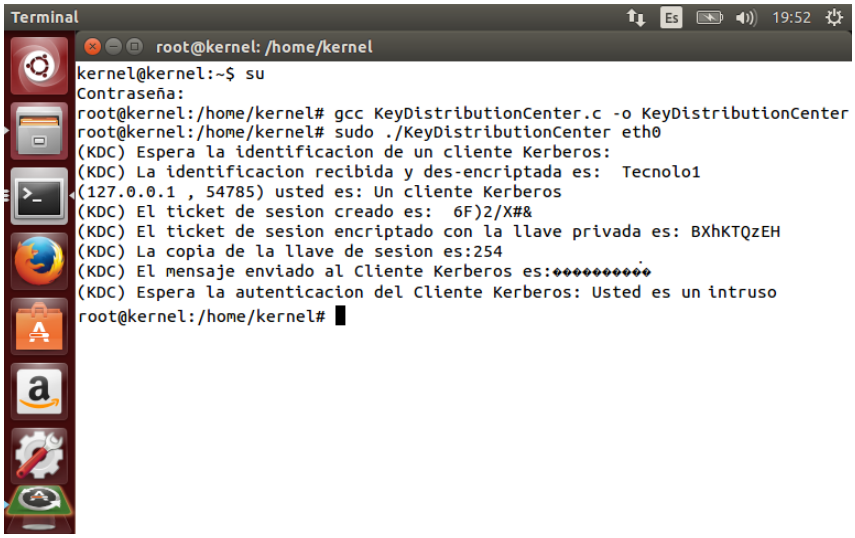

Figura 5.27: Respuesta del KDC al hombre en el medio, despues de recibir su autenticador.

En el siguiente caso se observa a un MITM que al no haber obtenido ni un ticket de sesión, ni de servicio, intenta hacerse pasar por un cliente Kerberos y se comunica con un cliente Kerberos, tal y como se observan en las Figuras 5.28 y 5.29.

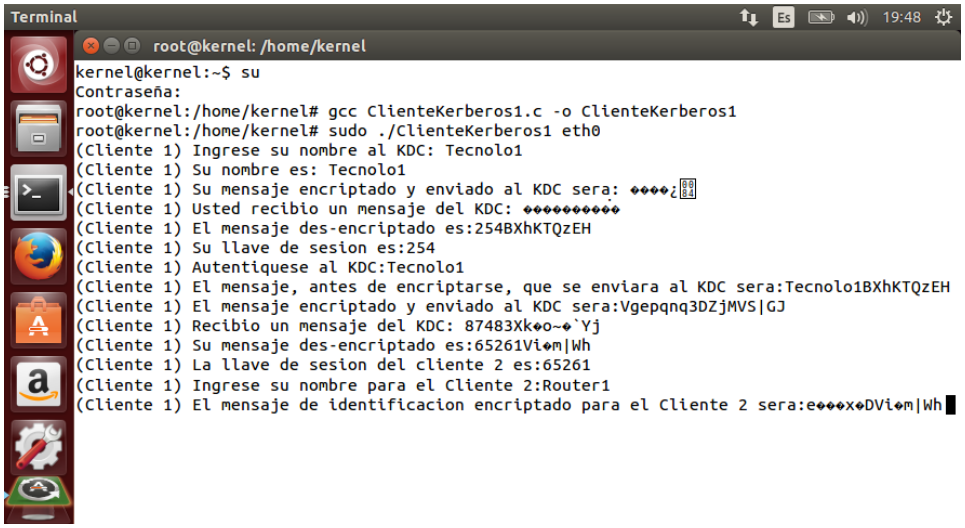

Figura 5.28: Hombre en el medio haciendose pasar por un cliente Kerberos.

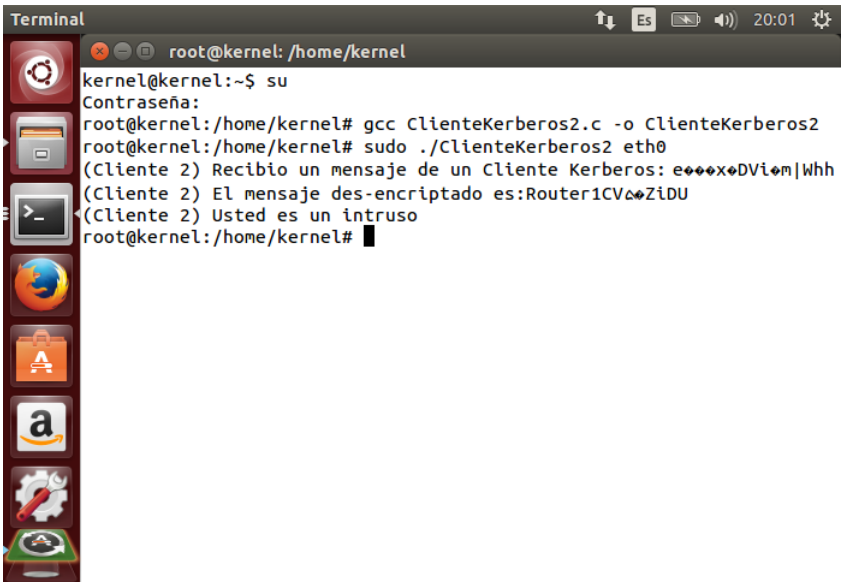

Figura 5.29: Respuesta del cliente Kerberos al hombre en el medio, despues de recibir su identificación.

Hemos concluido con el funcionamiento de la nueva propuesta, como puede observar, no existe forma de que un MITM pueda acceder a la información que se envía, y no puede hacerse pasar por un cliente Kerberos. Esta nueva propuesta, garantiza una mejora en la autenticación IPv6, siendo mucho más segura que el IPv6 convencional.

5.5. Probabilidad que un MITM logre obtener un ticket Kerberos

Hemos mencionado al inicio del trabajo que el objetivo de la propuesta es mejorar la autenticación de IPv6, y hacer de esta inmune al ataque de MITM. Esta propuesta se logró con el desarrollo de la propuesta presentada como "IPv6K". Sin embargo, todavía existe una mínima posibilidad que un MITM logre burlar la seguridad del KDC y de esta manera obtener un ticket autentico, lo que conllevaría a que pueda iniciar una comunicación Kerberos con un cliente Kerberos autentico.

De forma específica esta posibilidad es casi nula y considerando que cada ticket, ya sea de sesión o de servicio tiene un tiempo de vida y luego se debe realizar un nuevo proceso de identificación y autenticación con el KDC, las posibilidades que un MITM mantenga una comunicación bajo la propuesta planteada se reducen aún más. En este punto mostraremos cual es la probabilidad que un MITM logre burlar la seguridad del KDC para obtener un ticket.

- Primero un MITM debe lograr identificarse de manera satisfactoria con el KDC para obtener un ticket de sesión, para ello deberá obtener el User Name autentico de un cliente Kerberos registrado, el KDC verificará dicho el User Name presentado, la IP del cliente, la llave de sesión que pueda presentar, además de otros parámetros que el KDC ya tendrá antes de iniciar la sesión.

  Ahora existe una cantidad de bits determinada para un User Name en Kerberos, tal y como nos lo explican [Mic15, KN93], dicha cantidad es de 6 a 8 bits, y cada bit tiene un rango de caracteres posibles dicha cantidad de caracteres es de 950. Lo que conlleva a una construcción de un User Name utilizando uno de los 950 caracteres para cada bit. Finalmente la probabilidad mencionada seguirá la siguiente formula:

  $$Probabilidad = \frac{1}{x^y} * 100 \tag{5.1}$$

  Donde "x" es el rango posible de caracteres en un bit, en este caso 950.

  Y "y" es el número de bits que conforman el User Name, el cual como máximo es 8.

  Resolviendo la formula mencionada se obtiene que la probabilidad para que un MITM logre obtener el User Name correcto para presentarlo a un KDC, será de:

  $$1{,}507339769528e-22$$

Podemos concluir que la probabilidad de obtener el User Name de un cliente Kerberos es casi nula, y si adicionamos el hecho que un MITM no solo debería tener el User Name autentico, sino poder clonar la IP del cliente Kerberos real, así como su llave de sesión real y otros parámetros adicionales, conllevarían a una probabilidad aún más próxima a cero.

- Segundo el proceso anterior se realiza solo para obtener un ticket de sesión del KDC, el cual como se menciono tiene un tiempo de vida y expirado su tiempo de vida se debe repetir el proceso anterior.

  Luego un MITM no puede acceder a un ticket de sesión, tampoco puede usarlo para iniciar una comunicación con otros clientes, para ello necesita un ticket de servicio y para solicitarlo debe autenticarse al KDC y presentar el ticket de sesión autentico. Tal y como se observa en [For07, Tec09, KN93] los campos de un ticket Kerberos se encapsulan también en 8 bits. En resumen si un MITM desea obtener un ticket de servicio del KDC, la probabilidad seria de:

$$Probabilidad = \frac{1}{x^{2y}} * 100 \tag{5.2}$$

  Resultando en:

$$2{,}272073180803e - 46$$

  A este valor le adicionamos el hecho que el KDC verificara los campos del posible ticket de sesión que reciba con su base de datos, además de comprobar el autenticador recibido, resultando en un seguridad aun mayor y una probabilidad aún más próxima a cero, que la que se presentó.

  En conclusión las probabilidades que un MITM logre obtener un ticket de sesión o de servicio son casi nulas, existe una muy mínima posibilidad que pueda lograr burlarse al KDC, sin embargo dicha posibilidad es prácticamente imposible, además de considerar la seguridad del tiempo de vida de los tickets.

5.6. Comparación entre IPv6 e IPv6K

Si comparamos las características de la nueva propuesta, presentada como IPv6K, frente al IPv6, se presentara dos aspectos resaltantes, los cuales son el retardo de mensajes y la seguridad de los mensajes. Dado que otros aspectos tales como la tasa

de bits o capacidad, serán similares, pero no por ello no se deben mencionar.

Después de probar una comunicación entre dos clientes utilizando IPv6K, frente a IPv6, se obtuvo los siguientes resultados.

IPv6	IPv6K
A un rango de paquetes de 1536 a 65504 Bytes, se alcanzan los 94 Mbits/s en paquetes UDP.	A un rango de paquetes de 1536 a 65504 Bytes, se alcanzan los 97 Mbits/s en paquetes UDP.
A un rango de paquetes de 32 a 1536 Bytes, el porcentaje de paquetes perdidos va disminuyendo entre los intervalos de 0.35% a 0.15%.	A un rango de paquetes de 32 a 1536 Bytes, el porcentaje de paquetes perdidos va disminuyendo entre los intervalos de 0.25% a 0.1%.
A un rango de paquetes de 8192 a 65504 Bytes, el porcentaje de paquetes perdidos va disminuyendo entre los intervalos de 0.09% a 0.04%.	A un rango de paquetes de 8192 a 65504 Bytes, el porcentaje de paquetes perdidos va disminuyendo entre los intervalos de 0.05% a 0.01%.
A un rango de paquetes de 32 a 1536 Bytes, en una red local el retardo entre el paquete enviado y el mismo paquete en retornar es de 0.02 ms.	A un rango de paquetes de 32 a 1536 Bytes, en una red local el retardo entre el paquete enviado y el mismo paquete en retornar es de 0.14 ms.
A un rango de paquetes de 1536 a 65504 Bytes, en una red local es casi instantáneo.	A un rango de paquetes de 1536 a 65504 Bytes, en una red local es casi instantáneo.

Figura 5.30: Comparación entre IPv6 e IPv6K.

Finalmente las gráficas con los resultados específicos al momento de comparar IPv6 con la propuesta IPv6K, en un entorno de red local, son los siguientes:

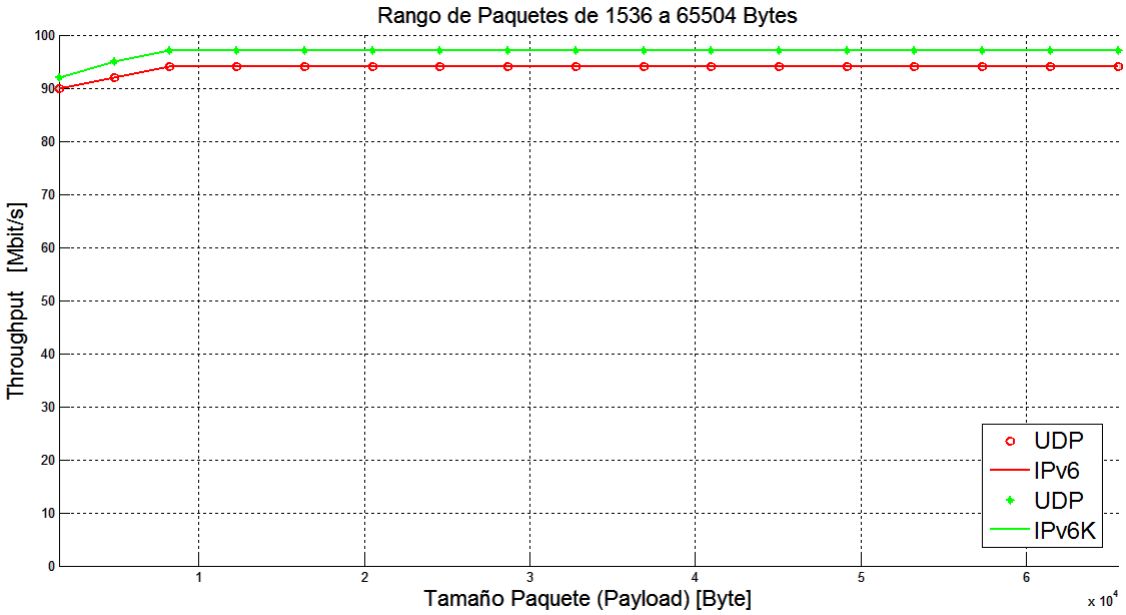

Figura 5.31: Cantidad de datos transferidos por segundo, en un rango de 1536 a 65504 bytes.

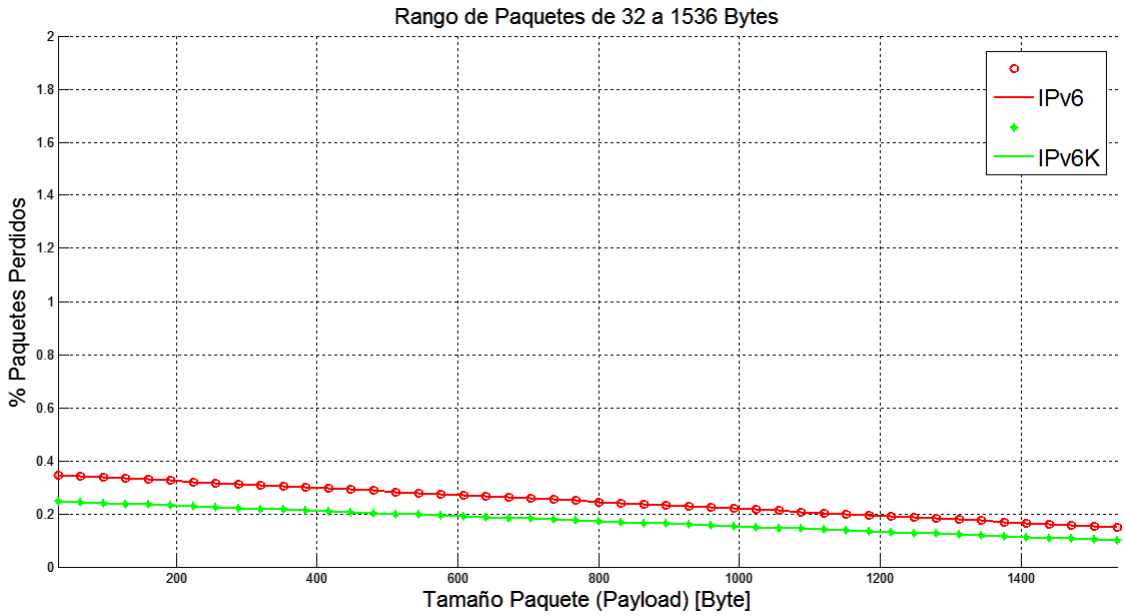

Figura 5.32: Porcentaje de paquetes perdidos, en un rango de 32 a 1536 bytes.

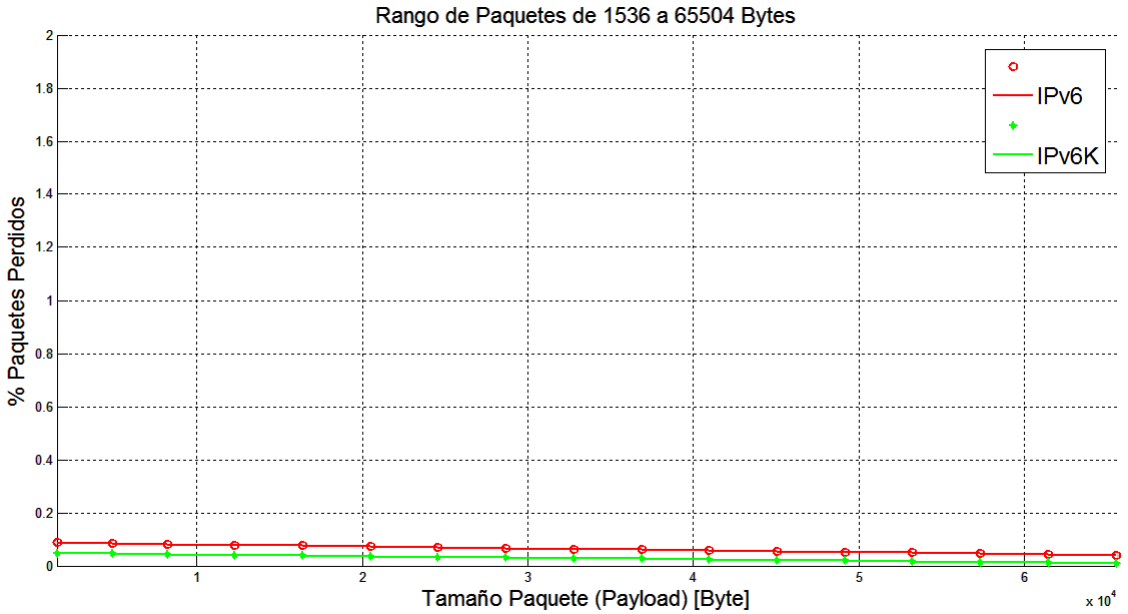

Figura 5.33: Porcentaje de paquetes perdidos, en un rango de 1536 a 65504 bytes.

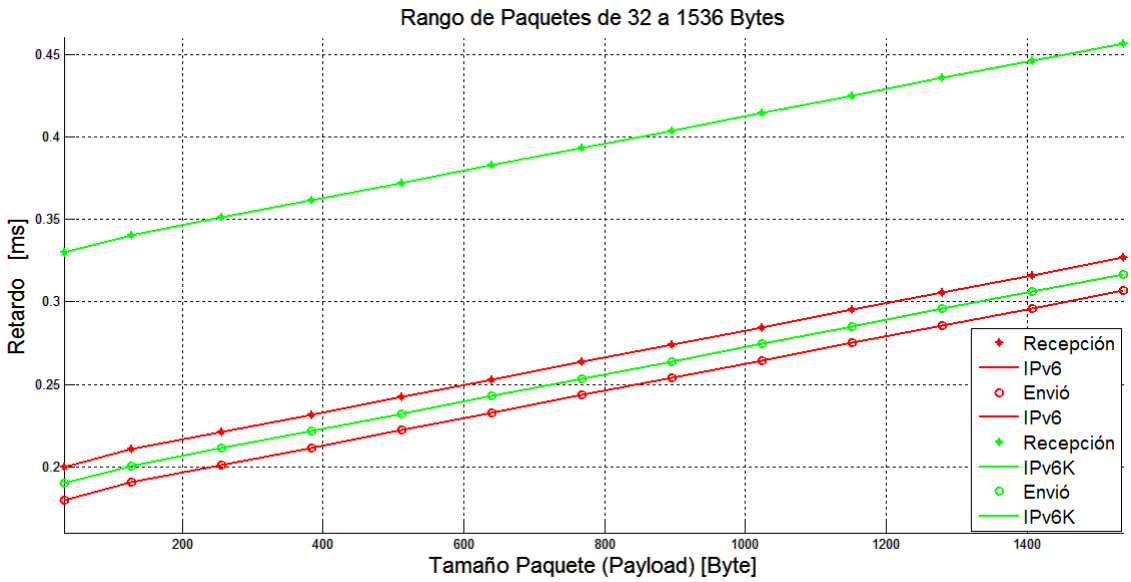

Figura 5.34: Retardo del trayecto de envió y recepción de paquetes, en un rango de 32 a 1536 bytes.

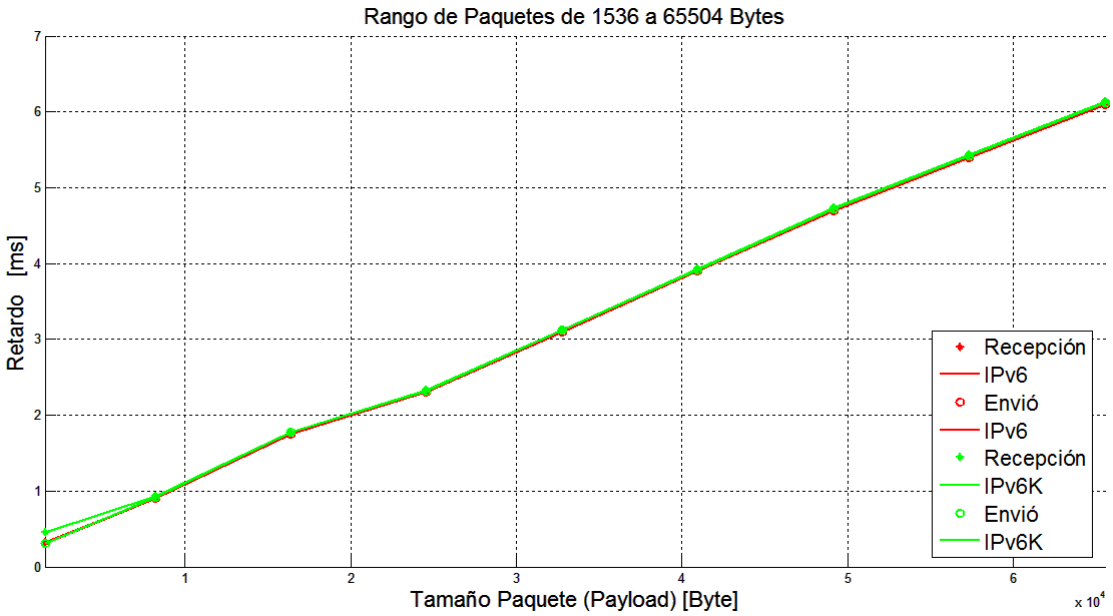

Figura 5.35: Retardo del trayecto de envió y recepción de paquetes, en un rango de 1536 a 65504 bytes.

6 Conclusiones y trabajos futuros

En este capítulo se presentaran las conclusiones del trabajo y se darán futuras líneas de investigación.

6.1. Conclusiones

- El planteamiento de utilizar el protocolo Kerberos junto con IPv6 garantiza mayor seguridad, debido a que el proceso de autenticación es más riguroso y se puede detectar la presencia de un MITM.
- Es importante que exista una etapa de logeo a Kerberos al inicio de la sesión para realizar la primera comunicación, consideramos que este cambio de la topología es necesario.
- De la misma forma que los passwords, debido a que también existen comunicaciones que se basan en el intercambio de certificados o credenciales, existe una verificación del certificado del servidor por parte del cliente y una verificación de las credenciales del cliente por parte del servidor, este planteamiento es muy riesgoso.
- La nueva propuesta presenta menor latencia a mayor cantidad de tasa de bytes enviados, incluso a mayores tasas de bytes es posible enviar los paquetes casi a la misma velocidad que lo realiza IPv6.
- A comunicaciones donde la tasa de bytes es pequeña, existe una pendiente que nos indica que la nueva propuesta presenta un retardo mayor en las comunicaciones.
- La seguridad en las redes es un factor que no solo depende de la autenticación, sino de la encriptación y de la integridad, los cuales son puntos que también deben mejorarse.
- Finalmente la nueva propuesta presentada es el inicio para poder seguir mejorando IPv6, dado que este protocolo es escalable y se puede adaptar.

6.2. Futuras líneas de investigación

A pesar de los esfuerzos de introducir una mejora en la forma de autenticación en IPv6, todavía resta un largo camino por recorrer. Hemos dejado en claro la

importancia de la autenticación, así los puntos que se deben solucionar con el tiempo. Un gran aporte que sería resaltante incluir llevar los datos no en el payload del paquete IPv6, sino en el campo de las opciones de IPv6.

Reconocimientos

Se me es muy grato agradecer a todas las personas que hicieron posible que este trabajo pueda ser concretado, a la Universidad Católica San Pablo y en especial a la Escuela Profesional de Ingeniería Electrónica y de Telecomunicaciones. Al hacer una mira atrás vienen muchos nombres a mi mente, compañeros de estudios, jurados de tesis y docentes de la Universidad Católica San Pablo que me motivaron a concluir esta tesis. A mi familia que me impulso a seguir y por ser un ejemplo para mi. En especial a mi asesor el Mag. Julio Omar Santisteban Pablo, por su amabilidad, constante motivación, asesoría y guía en todo este proceso.

Bibliografía

[AD00] M. Ali and J. Deogun. Power-efficient design of multicast wavelength-routed networks. *IEEE Journal on Selected Areas in Communitations*, 18(10): 852–862, 2000.

[Alo12] Chema Alonso. Fc00::1 (algunos) ataques en ipv6. *NCN*, November 3, 2012.

[Cal] E. Caliskan. Ipv6 transition and security threat report. *2014. http://www.ccdcoe.org/publications/articles/IPv6-Report.pdf.*

[Cas16] Y. Castillo. Agotamiendo ipv4 en la region de latinoamericana. *Prisma, Volumen 5, paginas de 26 al 28*, 2016.

[CB10] J. Castro and S. Benito. Ipsec. October 5th of 2010.

[DH95] S. Deering and R. Hinden. Rfc 1883: Internet protocol, version 6 (ipv6). December 1995.

[dSODFU] Grupo de Sistemas Operativos DATSI FI UPM. Protocolo ipsec.

[For07] Internet Engineering Task Force. Kerberos ticket. *IETF*, 2007.

[Fra09] Hugo Adrian Francisconi. *IPsec en Ambientes IPv4 e IPv6.* Primera Edicion, Diciembre 2009.

[Gar12a] C. Garcia. Analisis de seguridad en redes ipv6. July 2012.

[Gar12b] C. Garcia. Analisis de seguridad en redes ipv6. July 2012.

[Hat05] Red Hat. *Manual de referencia.* Red Hat Enterprise Linux 4, 2005.

[IPv00] IPv6Mx. Ipv6 cambia la seguridad: Â¿estÃ¡ listo su negocio? *Network Information Center MÃ©xico S.C.*, 2000.

[KB] H. Krawczyk and M. Bellare. Hmac: Keyed-hashing for message authentication. *February 1997. https://tools.ietf.org/html/rfc2104.*

[KN93] J. Kohl and C. Neuman. The kerberos network authentication service (v5). *RFC1510*, September 1993.

[KS12] D. Kuegler and Y. Sheffer. Password authenticated connection establishment with the internet key exchange protocol version 2 (ikev2). *IETF*, June 2012.

[LB08] Leuven and Belgium. *Privacy Enhacing Technologies.* 8th International Symposium, PETS 2008, 2008.

[McC13] Daryl McCartney. An introduction to kerberos in os x. May 21, 2013.

[Mex16] NIC Mexico. Agotamiento ipv4. *lacnic*, Consultado Noviembre, 2016.

[Mic15] Support Microsoft. Problems with kerberos authentication when a user belongs to many groups. *Microsoft*, July 14, 2015.

[MS98] D. Maughan and M. Scherler. *Internet Security Association and Key Management Protocol (ISAKMP).* RFC 2408, November 1998.

[Pat13] Eleven Path. Ataque de hombre en el medio (mitm). *Evil Foca*, 9th of August of 2013.

[Pea14] Mervin Pearce. Kerberos authentication. *Security Audit and Control Solutions*, Jan 21, 2014.

[Pos81] J. Postel. Rfc 791: Internet protocol. September 1981.

[Pro05] Cisco Networking Academy Program. *Fundamentos de Seguridad de Redes.* Cisco Systems, Inc., 2005.

[Rfc] Rfc2460. Internet protocol, version 6 (ipv6) specification.

[Ros16] Alvaro Rodrigo Reyes Rosado. Ataques en redes de datos ipv4 e ipv6. *Escuela Tecnica Superior de Ingenieria InformÃ¡tica Malaga*, Noviembre de 2016.

[Tec] Technet. Hashed message authentication code functions. *http://technet.microsoft.com/en-us/library/cc962016.aspx.*

[Tec09] TechNet. How the kerberos version 5 authentication protocol works. *Microsoft*, October 7, 2009.

[TEC14] HUAWEI TECHNOLOGIES. Ipv6 technical white paper. February 19th of 2014.

[Var13] JoaquÃn RodrÃguez Varela. Defcon: Ataques al protocolo ipv6. *DEFCON*, August 6th of 2013.

[Wol] M. Wolfe. What is hmac authentication and why is it useful? *October 20, 2012. http://www.wolfe.id.au/2012/10/20/what-is-hmac-and-why-is-it-useful.*

[Yan14] X. Yang. Migrating industrial ipv4 network to ipv6. March 2014.

[yFR13] C. Medina y F. RodrÃguez. *Caracterizacion de IPv6.* Volumen 17, paginas de 111 al 128, 2013.

[yRSA16] J.R. Gomez Rodriguez y R. Sandoval ArÃ©chiga. Ipv6 el tiempo ha llegado. *Universidad Autonoma de Zacatecas, Unidad AcadÃ©mica de Ingenieria Electrica*, 2016.

Nomenclatura

DNS	Denial of Service
ESP	Encapsulating Security Payload
HMAC	Hash Message Authentication Code
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISAKMP	Internet Security Association and Key Management Protocol
MITM	Man in The Middle
NDP	Neighbor Discovery Protocol
QoS	Quality of Service
SLAAC	Stateless Address Autoconfiguration